



Universidad Complutense de Madrid

Facultad De Informática
Trabajo de Fin de Grado

IS CASE TOOL - DOCS
Daniel Reyes Tubón

Director
Antonio Sarasa Cabezuelo

Madrid-2018

Resumen

Para este TFG se ha creado una herramienta case para mantener, gestionar y generar documentos referentes a la asignatura de Ingeniería del Software.

El objetivo del trabajo es proveer una herramienta de ayuda a estudiantes y profesores en el contexto de la asignatura de **IS**; permitiendo asignar proyectos documentales a cada equipo de trabajo y facilitando a los estudiantes el relleno y generación de los distintos documentos requeridos en la asignatura como son **SRS y Plan de Proyecto**.

Para ello, se ha desarrollado una herramienta web, que mediante autenticación diferencia dos escenarios de acuerdo con el tipo de usuario, utilizando para ello una interfaz común.

El primero se dedica a los Profesores (Administradores) quienes, tienen la posibilidad de gestionar de forma básica los Equipos de trabajo en función de los Estudiantes previamente registrados. Para ello, los profesores deben ser dados de alta en el sistema previa puesta en producción de la herramienta, funcionalidad que también está disponible a través de la interfaz web mediante el cambio de un simple 'flag' en el fichero de configuración del proyecto.

Mientras que, el segundo escenario se dedica a los Estudiantes (Miembros) quienes, mediante una sencilla interfaz podrán editar, guardar y generar los documentos de su Equipo; recibiendo para ello consejos y ejemplos de ayuda en el relleno de cada uno de los apartados de un documento.

Abstract

A Case Tool has been created in this work to maintain, manage and generate documents related to the Software Engineering course.

The work's purpose is to provide a tool to help students and teachers in the context of the IS course, allowing the assignment of documentary projects to each work team and facilitating the students the completion and generation of the different documents required in the subject such as SRS and Project Plan.

To this end, a web tool has been developed, which uses authentication to differentiate two scenarios according to the type of user, using a common interface.

The first one is dedicated to the Teachers (Administrators) who have the possibility to manage in a simple way the Work Teams according to the previously registered Students. For this, teachers must be registered in the system before the tool is put into production, a feature to do it is also available through the web interface by changing a simple 'flag' in the project configuration file.

The second scenario is dedicated to the Students (Members) who, through a simple interface, will be able to edit, save and generate the documents of their Team, receiving advices and examples to help in the completion of each document's section.

Agradecimientos

Mis más sinceros agradecimientos, van dedicados en primera instancia a mi director de proyecto, Antonio Sarasa Cabezuelo, por el apoyo y tiempo dedicados a trabajar en la realización. Siempre valoraré sus enseñanzas y disposición para ayudar cuando ha sido necesario.

Agradezco a mi madre y seres más allegados, quienes siempre estuvieron apoyándome en la culminación de mi trabajo, sin importar las circunstancias.

Además, dedico un agradecimiento especial, a la dirección de la empresa Provide HCM People, en la actualmente trabajo ya que; sin su formación, apoyo y flexibilidad, este trabajo no habría sido posible.

De igual manera, de antemano agradezco a todos y cada uno de los lectores, por la atención prestada.

Índice general

| | |
|--|----|
| Resumen | 2 |
| Abstract | 3 |
| Agradecimientos | 4 |
| Índice general | 5 |
| 1. Introducción | 6 |
| 1.1 Introducción al TFG | 6 |
| 1.2 Contexto del proyecto | 6 |
| 1.3 Estado del arte | 7 |
| 1.4 Estructura de la memoria | 9 |
| 1. Introduction | 10 |
| 1.2 Context of the project | 10 |
| 1.3 State of art | 11 |
| 1.4 Structure of the Report | 13 |
| 2. Objetivos Generales | 14 |
| 2. 1 Objetivos | 14 |
| 2.2 Motivación | 15 |
| 3. Especificación de requisitos | 16 |
| 3.1 Casos de Uso | 16 |
| 3.2 Características del Usuario | 16 |
| 3.3 Requisitos Funcionales | 17 |
| 4. Tecnologías aplicadas | 31 |
| 5. Arquitectura y modelo de datos | 33 |
| 5.1 Arquitectura de la herramienta | 33 |
| 5.2 Arquitectura de la Implementación | 36 |
| 5.3 Modelo de datos | 37 |
| 6. Diseño e Implementación | 42 |
| 6.1 Principios de diseño | 42 |
| 6.2 Interfaz de Usuario e Implementación | 43 |
| 6.2.1 Visión Estructural del Código Implementado | 43 |
| 6.2.2 Una Herramienta con Información Autogenerada | 44 |
| 6.2.3 Interfaz de Usuario y su Funcionalidad Implementada | 49 |
| 6.2.3.1 UI y Funcionalidades para Miembro Administrador | 49 |
| 6.2.3.2 UI y Funcionalidades para Miembro (No Administrador) | 58 |
| 6.2.3.3 UI y Funcionalidades Generales (Todos los Usuarios) | 68 |
| 7. Conclusiones y Trabajos de futuro | 72 |
| 7.1 Conclusiones del Trabajo | 72 |
| 7.2 Trabajos de futuro | 72 |
| 7. Conclusions and future works | 73 |
| 7.1 Conclusions | 73 |
| 7.2 Future work | 74 |
| 8. Contribuciones | 74 |
| Bibliografía | 75 |
| ANEXOS | 77 |
| APÉNDICE A. Guía de instalación de la herramienta | 77 |
| APÉNDICE B. Manual de usuario | 83 |

1. Introducción

1.1 Introducción al TFG

Este trabajo se formula como apoyo a la enseñanza de la asignatura de **Ingeniería del Software** con la intención de facilitar el entendimiento y aprendizaje de conocimientos avanzados y necesarios para el desarrollo de un Proyecto y su Documentación, orientando a los estudiantes poco experimentados a la adquisición de dichos conocimientos mediante la ayuda y ejemplificación del desarrollo del documento de especificación de Requisitos '**SRS**'.

A través de una herramienta web, "**IS Case Tool - Docs**", este proyecto permite a un equipo de estudiantes generar un documento común y autogestionado que, mediante consejos y ejemplos mostrados sobre una Interfaz de Usuario muy intuitiva, guía a los autores en la construcción del mismo. Para ello, la herramienta ofrece al Profesor (Administrador) la posibilidad de dar de alta a Equipos de Trabajo, generando así un documento único, al que pueden acceder, editar y guardar todos los miembros del equipo.

IS Case Tool - Docs, es capaz de generar todos los apartados (a rellenar) y guías de ayuda para fabricar un documento **SRS** a partir de una plantilla preconfigurada y basada actualmente en "**IEEE Recommended Practice for Software Requirements Specifications Std. 830-1998**" ofreciendo las siguientes funcionalidades implementadas y que a posteriori se explican con detalle:

- Funcionalidades Generales:
 - Registro de usuario en el Sistema.
 - Acceso a la Herramienta mediante autenticación.
 - Consulta del Perfil de Usuario.
 - Modificación del Perfil de Usuario.
- Funcionalidades del Usuario Administrador:
 - Alta de Equipos de Trabajo.
 - Modificación de Miembros de un Equipo de Trabajo.
 - Baja de un Equipo de Trabajo.
- Funcionalidades del Usuario Miembro:
 - Acceso al documento de su equipo.
 - Modificación de todos y cada uno de los apartados del Documento (incluye Auto Guardado de la información).
 - Previsualización del Documento PDF.
 - Generación del Documento en formato PDF.

1.2 Contexto del Proyecto

Cada año, en la Facultad de Informática, se imparte la asignatura de "Ingeniería del Software", en la que se propone a los estudiantes desarrollar un proyecto en equipo en el que, durante la primera etapa, se requiere la realización de toda la Documentación incluyendo los documentos de SRS (especificación de Requisitos), Plan de Proyecto y Memoria Final.

Dicha tarea, constituye uno de los mayores retos de la asignatura, sobre todo para los estudiantes poco experimentados en el desarrollo de proyectos de esta índole.

En dicho contexto, este proyecto, expone el prototipo de la herramienta **IS Case Tool – Docs**, pensada para la organización y construcción guiada de dichos documentos en un entorno compartido por un mismo equipo de trabajo; es esta ocasión concretamente sobre el **SRS**.

Sin embargo, y aunque este trabajo se enmarca en un único caso, en realidad el sistema, ha sido preparado, para poder continuar en un futuro con la generación del resto de Documentos e incluso una gestión de equipos más extendida.

1.3 Estado del Arte

Hoy en Día, el trabajo cooperativo está en auge en todos los ámbitos, desde educacionales, comenzando su aplicación desde educación infantil y en todos los niveles posteriores, hasta profesionales, considerando casi cualquier área. Se escucha hablar de equipos científicos, técnicos, de investigación o de diseño, aplicados a campos tan distintos como la medicina, la ingeniería, la política o la seguridad estatal.

Se entiende por trabajo en equipo una labor realizada por varios individuos, donde cada uno efectúa una parte en pos de un objetivo común. Para que dicha obra se considere trabajo en equipo, éste debe tener una estructura organizativa y cada miembro un rol definido, obteniéndose, en general, muy buenos resultados. [22]. Estos aspectos se aplican directamente en la realización de proyectos de Ingeniería del Software, campo que enmarca el presente trabajo fin de grado.

Las primeras fases de los proyectos software se basan en la generación de documentación, que aglutina el diseño del sistema a construir, los requisitos a cubrir y las fases a completar, entre otros aspectos importantes para finalizar con éxito el proyecto a realizar. Es muy importante que todos los miembros del equipo tengan acceso a los documentos generados durante todo el proceso para su consulta y completitud, lo que sugiere la idea de utilizar alguna herramienta o almacenamiento en la nube. Por otro lado, si puede facilitarse la tarea asociada a producción de los documentos, se simplifica significativamente la fase de diseño y se disminuyen los posibles errores que se puedan cometer.

Se pueden encontrar diversas herramientas en el mercado, tanto gratuitas como de pago, orientadas a la gestión y generación documental. A continuación, se presentan algunas de las mismas, citando sus características más importantes, así como sus carencias o puntos débiles.

- **Google Docs** es la primera herramienta que se ha analizado. Se trata de una herramienta gratuita y multiplataforma propuesta por Google Inc., cuyo lema es *“Nuestra misión es organizar la información del mundo para que todos puedan acceder a ella y usarla”* [23]. Acorde con dicha idea, Google Docs soporta el almacenamiento en la nube de documentos y su edición, apostando por la movilidad. Proporciona una plataforma que permite compartir documentos entre varios participantes, modificaciones en tiempo real y la posibilidad de incluir comentarios sobre los documentos gestionados. Ofrece la posibilidad de realizar conversaciones en línea entre miembros de un equipo de trabajo, tanto escritas como de voz o vídeo. Además, se han desarrollado diversos complementos que añaden funcionalidades específicas, como la posibilidad de realizar diagramas, crear plantillas, gestionar la biblioteca o corregir ortografía y gramática, entre muchas otras. [24] Su carencia más importante considerando el ámbito de aplicación objetivo del proyecto es la no existencia de un complemento suficientemente robusto y completo para dar soporte al diseño de proyectos.
- La siguiente herramienta que ha resultado interesante revisar ha sido **Heflo**, desarrollada por Venki. Orientada a la documentación de procesos de negocio, es una herramienta de pago que ofrece

almacenamiento en la nube y facilidad para compartir diagramas y documentación con los participantes en un proyecto. Se basa en notación BPMN (Business Process Model and Notation por sus siglas en inglés), estandarizada y gráfica, cualidades que facilitan la comprensión de los procesos descritos de esta forma [25]. Es capaz de generar documentación a partir de las gráficas realizadas con un formato limpio, y permite descargar imágenes y realizar control de versiones, así como exportar o importar modelos en formato BPMN. Presenta diferentes planes de contratación, ofreciendo funcionalidades como listados de tareas, alarmas sobre plazos de entrega, integración con Slack como herramienta para conversaciones en línea o herramientas de monitorización [26]. Se trata de una opción potente a la vez que cara, muy orientada a empresa, presentando la paradoja de ofrecer complementos que no son necesarios en la fase de diseño de proyectos y no cubrir todas las necesidades objetivo del desarrollo realizado.

- **REM** es una herramienta gratuita desarrollada por el departamento de lenguajes y sistemas de la Universidad de Sevilla, orientada principalmente a la gestión de requisitos, siguiendo la metodología desarrollada por D. Amador Durán. Actualmente se ofrece una versión experimental de la herramienta, y aunque sencilla, cubre todas las necesidades de la fase de ingeniería de requisitos de un proyecto [27]. Su principal carencia es la dificultad de compartir información en tiempo real entre miembros de un equipo que se no se encuentren reunidos físicamente, ya que no permite almacenamiento en la nube.
- Una opción muy completa es **Enterprise Architect, de SparxSystems**. Es capaz de cubrir todas las etapas de diseño y modelado para diversas áreas, desde procesos de negocio hasta desarrollo software, incluyendo proyecto orientados a sistemas embebidos, y cubriendo también diseño de sistemas o de bases de datos. Ofrece la posibilidad de realizar y componer distintos tipos de gráficos, entre los que destaca la notación UML para casos de uso. A partir de los requisitos y diseños introducidos en la herramienta, genera documentación e informes. Una de las características más atractivas es la generación a partir de los modelos realizados de código fuente en varios lenguajes de programación (por ejemplo, C#, Java, PHP, VHDL o Python, entre otros). En conjunto, se trata de una opción muy potente, que permite un alto grado de cooperación entre los integrantes de un equipo de trabajo, asignando roles a los participantes e integrando control de versiones, y brindando la posibilidad de ser desplegada sobre un servidor (basado en la nube o en red de área local) o servicio de almacenamiento en la nube. Como ya se ha adelantado, cubre desde el modelado hasta el análisis del impacto producido por cambios en el diseño, ofrece modelos y plantillas tanto para modelado como para documentación e informes, integra varias herramientas para dar soporte a cada fase de un proyecto y está basado en estándares como UML, BPMN y SysML [28]. Con todo ello, se trata de una alternativa robusta y eficaz, a la vez que cara, ofreciendo diversos planes de suscripción para su uso.
- Centrándose únicamente en la gestión documental, Ibermática ofrece la plataforma **iberDok**, para la generación, gestión y edición de documentación a partir de plantillas. Permite la utilización de objetos dinámicos y la producción de documentos inteligentes basados en reglas y dependencias. Orientado a empresas que generan y distribuyen por diversas vías (correo electrónico, impresión en papel o publicación en formato electrónico) grandes volúmenes de documentos [29], podría considerarse como una opción para la generación de documentos basados en plantillas unificadas, pero no ofrece funcionalidades básicas para el ámbito al que se orienta este proyecto.

- Dentro de la misma familia de herramientas orientadas a la gestión documental, cabe mencionar **openKm**, que integra control de versiones y permite controlar el ciclo de vida de los distintos documentos. Al tratarse de un repositorio en la nube, permite la colaboración entre miembros de un equipo para la generación de documentación, pero no está orientada de forma específica a diseño de sistemas o desarrollos en marcados en el área IT. Aunque admite el diseño de flujos de trabajo, no ofrece herramientas específicas para modelar gráficamente casos de uso o diagramas de proceso. Se trata de una opción adecuada para la administración de documentos, que permite definir tareas automáticas y el uso de distintos módulos que aumentan las funciones básicas ofertadas, según necesidades más específicas, como puede ser la firma digital o la gestión de tareas. [30]

Como puede inferirse de todo lo expuesto anteriormente, existen en el mercado una gran variedad de herramientas, plataformas y aplicaciones orientadas a la gestión, generación y almacenaje de documentación. Con el presente proyecto, se pretende desarrollar una herramienta dinámica, orientada específicamente a la gestión de requisitos en el ámbito de Ingeniería del software, que permita a los distintos equipos desarrollar documentos de forma colaborativa, respetando los roles asignados y facilitando la labor que conlleva dicha fase de diseño software.

1.4 Estructura de la Memoria

En el transcurso de la lectura, se puede apreciar en el documento una estructura organizada en capítulos.

El Capítulo 1, ofrece una introducción al trabajo realizado, exponiendo los aspectos fundamentales para ubicar al lector en un contexto adecuado, pretendiendo así, cimentar una idea general del tema principal sobre el que fundamenta este trabajo, incluyendo funcionalidades y símiles del mismo.

El Capítulo 2, describe las razones que han motivado la realización del trabajo y los objetivos generales de su desarrollo; los cuales, se han generado a partir del estudio de las necesidades en el apoyo educacional para con los profesores y estudiantes de generaciones futuras. Así también, desde una perspectiva tecnológica, se pretende exponer algunos aspectos fundamentales del desarrollo web en la actualidad.

El Capítulo 3, presenta la especificación de requisitos del sistema, estructurando y definiendo de forma concisa las funcionalidades que se han desarrollado.

El Capítulo 4, expone las distintas tecnologías utilizadas para el desarrollo, describiendo sus características y motivos de elección.

El Capítulo 5, explica la Arquitectura utilizada como base del desarrollo y describe el Modelo de Datos utilizado para gestionar y mantener la información que permite al sistema ofrecer toda su funcionalidad.

El Capítulo 6, expone y explica todas las decisiones de diseño que se han tomado para el sistema y los principios aplicados para tomarlas. Además, incluye un análisis detallado de la implementación.

El Capítulo 7, cierra el conjunto presentando las conclusiones a las que se ha llegado después de la realización del trabajo y propone posibles extensiones que den completitud al mismo.

Finalmente, existe un apartado de anexos, donde se puede encontrar una guía de instalación de la herramienta y un manual de usuario básico, para el uso de **IS Case Tool – Docs**.

1. Introduction

1.1 Introduction to TFG

This work is meant to be a support one of to the main subject in Computer Science Engineering Degree as it is “Software Engineering “, so this way it will be easier to learn and understand basic and advanced knowledge for developing a project and write its entire documentation of Software Requirements Specification (SRS). Most of them for inexperienced students who will need a little help with the first steps to achieve this process of knowledge.

With a web application “IS Case Tool - Docs”, this project allows a group of students to generate a common and self-managed document that through some tips and examples shown on a very intuitive User Interface it guides to authors in the construction of it. So that, this tool offers the professor (user administrator) the possibility of registering work teams in a single and simple document which can be accessed, edited and saved by all team members.

IS Case Tool - Docs, is capable of generate all the sections (to be filled) and help guides to make an SRS document from a template which is preconfigured and currently based on “IEEE Recommended Practice for Software Requirements specifications Std. 830-1998”. This tool offers the following functionalities which are implemented and explained in detail afterwards.

General Functionalities:

- Registration of the user in the system
- Access to the Tool by authentication
- Check of the user profile
- Edition of the user profile.

Functionality of the administrator user:

- Registration of work teams
- Edition of member of a work team
- Removal of a work team

Functionality of the member user:

- Access to his team’s document
- Edition of each and every one of the sections of the document (with auto saved information)
- Preview of the PDF document
- Generation of the document in PDF format

1.2 Context of the Project

Every year, in the Faculty of Computer Science, the subject “Software Engineering” is taught, and students are proposed to develop a project in work teams in which, during the first stage it is required to accomplish all the documentation; including SRS document, Project plan and the Final Report.

This task is one the major challenges of the above-mentioned subject, especially for students with lack of experience in developing-project field.

In this context, this project has a prototype of the tool “IS Case Tool - Docs” designed for the organization and guided construction of these documents in an environment shared by the same team. Specially in this case it is about the SRS.

Although this work is part of a single case, the system itself. That has been prepared to be continued in the future with the generation of the rest of the documents and even more extended work teams.

1.3 State of art

Today, cooperative work is booming in all areas, from educational to professional areas, starting its application from early childhood education to all subsequent levels. We hear about scientific, technical, research or design teams which applied to different fields like medicine, engineering, politics or state security.

Teamwork is understood as a work carried out by several individuals, where everyone performs a small part in pursuit of a common goal. If we want this work to be considered teamwork, it must have an organizational structure and each member must have a defined role. This way we can obtain very good result in general [22].

These aspects are directly applied in developing software engineering projects, which is the main topic the actual Bachelor Thesis.

The first phases of software projects are based on generating documentation which brings together the design of the system to be built, the requirements to be covered and the phases to be completed, among other important aspects to successfully complete the project. It is very important that all members of the team have access to the documents that was generated during the whole process in order to these documents can be edited and completed properly, which suggests the idea of using a tool or storage in the cloud. On the other hand, if the task associated with generating documents can be facilitated, the design phase will be significantly simplified, and the possible errors will decrease as well.

We can find many tools in the market, both free and paid ones, aimed at the management and generation of documents.

Just below, some of them are presented with their most important characteristics, as well as their shortcomings or weak points.

- **Google docs:** it is the first tool that has been analyzed. It is a free and multiplatform tool proposed by Google Inc., whose motto is “Our mission is to organize the world’s information so that everyone can access it and use it” [23].
According to this idea, Google docs supports the storage in the cloud of documents and their edition, betting on mobility. It provides a platform that allows to share documents among several participants, real-time modifications and the possibility of including comments on the managed documents. It offers the possibility of conducting online conversation between members of a team, by writing, voice or video play. In addition, various add-ons have been developed that add specific functionalities such the possibility of making diagrams, templates, or managing the library or correcting spelling and grammar, among many other things. [24] Its most import lack, considering the objective scope of the actual project, it is the non-existence of a sufficiently robust and complement that supports the design of project.
- **Heflo:** it is developed by Venki and oriented to the documentation of business process. It is a payment tool that offers storage in the cloud and easy to share diagrams and documentation with the participants in a specific project. It is based on BPMN notation (Business Process Model and

Notation), standardized and graphic, these qualities facilitate the understanding of the process described in this way [25]. It is able to generate documentation from the graphics made with version control, as well as export or import models in BPMN format. It presents different hiring plans, offering functionalities such as task lists, alarms to deadlines, integration with Slack as a tool for online conversations or monitoring tools [26]. It is a powerful option and at the same time expensive, very oriented to company model, it also presents the paradox of offering add-ons that are not necessary in the design phase of projects and that are not covering all the objective that is needed of the development carried out.

- **REM:** it is a free tool developed by the Department of Languages and Systems of the University of Seville, mainly oriented to the management of requirements, it follows the methodology developed by Mr. Amador Durán. Currently, it offers an experimental version of the tool, and although it is simple, it covers all the requirements of the engineering phase of a project's requirements [27]. Its main lack is the difficulty of sharing information in real time between members of a specific team that are not physically reunited, since it does not allow storage in the cloud.
- A very complete option is **Enterprise Architect**, developed by SparxSystems. It is capable of covering all stages of design and modeling for various areas, from business processes to software development, including project oriented to embedded systems, and also covering systems or database design. It offers the possibility of making and composing different types of graphics, including UML notation for use cases. From the requirements and designs introduced in the tool, it generates documentation and reports. One of the most attractive features is the generation from models made of source code in various programming languages (for example, C#, Java, PHP, VHDL or Python, among others). Overall, this is a very powerful option, which allows a high degree of cooperation between members of a work team, assigning roles to participants and integrating version control, and providing the possibility of being deployed on a server (based on the cloud or local area network) or cloud storage service. As already mentioned, it covers from modeling to the analysis of the impact produced by design changes, offers models and templates for both modeling and documentation and reports, integrates several tools to support each phase of a project and is based on standards such as UML, BPMN and SysML [28]. With all this, it is a robust and effective alternative, while expensive, offering various subscription plans for use.
- Focusing exclusively on document management, **Ibermática offers the iberDok** platform, for the generation, management and editing of documentation from templates. It allows the use of dynamic objects and the production of intelligent documents based on rules and dependencies. Aimed at companies that generate and distribute large volumes of documents by different channels (e-mail, printing or publication in electronic format) [29], it could be considered as an option for the generation of documents based on unified templates, but it does not offer basic functionalities for the field targeted by this project.
- Within the same family of tools oriented to document management, it is worth mentioning **openKm**, which integrates version control and makes it possible to control the life cycle of the different documents. As it is a repository in the cloud, it allows collaboration between members of a team for the generation of documentation, but it is not specifically oriented to system design or developments in the IT area. Although it supports workflow design, it does not offer specific tools to graphically model usage cases or process diagrams. This is a suitable option for document management, which allows defining automatic tasks and the use of different modules that increase the basic functions offered, according to more specific needs, such as digital signature or task management. [30]

As can be inferred from the above, there is a wide variety of tools, platforms and applications in the market oriented to the management, generation and storage of documentation. With this project, it is intended to develop a dynamic tool, specifically oriented to the management of requirements in the field of software engineering, which allows the different teams to develop documents collaboratively, respecting the roles assigned and facilitating the work involved in this phase of software design.

1.4 Structure of the report

In the course of reading, you can see in this document a structure organized into chapters.

Chapter 1 offers an introduction to the work done, and it shows the fundamental aspects to place the reader in an appropriate context. It tries to build a general idea of the main topic which this work is based on, including functionalities and similes of it.

Chapter 2, describes the reasons that have motivated the work and the general objectives of its development; which have been generated from the study of the needs in support the education for teachers and students of future generations. Also, from a technological perspective, it is intended to expose some fundamental aspects of web development at present.

Chapter 3, presents the specification of system requirements, it structures and defines concisely the functionalities that have been developed.

Chapter 4, shows the different technologies used for development, it describes the characteristics and the reasons of choice.

Chapter 5, explains the Architecture used as the basis of the development and it describes the Data Model used to manage and maintain the information that allows the system to offer all its functionality.

Chapter 6, shows and explains all the design decisions that have been made for the system and the principles applied, indeed, to take them. In addition, it includes a detailed analysis of the implementation.

Chapter 7, closes the set of topics and it presents the conclusions reached after the completion of the work and it proposes possible extensions, so they can give completeness to it.

Finally, there is a section of annexes, where you can find a guide to install the tool and a basic user manual in order to use IS Case Tool - Docs.

2. Objetivos Generales

2.1 Objetivos

Como principal objetivo, el trabajo plantea el desarrollo de una herramienta case, **IS Case Tool – Docs**, que sirva de apoyo a estudiantes y profesores en el aprendizaje y enseñanza del cómo construir la documentación requerida en la asignatura de **Ingeniería del Software**. Pretendiendo así, facilitar la construcción cooperativa de un documento en un equipo de trabajo, a través de guías y ejemplos concretos, que se ponen a disposición de los estudiantes durante el uso de la herramienta.

En segunda instancia y a pesar de exponer el prototipo de una herramienta relativamente sencilla. Existe un trasfondo que lleva un objetivo claro, enfocado en las tecnologías utilizadas. Pues desde el punto de vista vocacional, hacia la Ingeniería Informática, estoy convencido de que, en este campo, una de las cosas más importantes es buscar siempre soluciones e ideas que den valor a nuestro trabajo.

Por tal motivo y hablando más concretamente; se ha enfocado el desarrollo de este trabajo, dentro del mundo **SAP [1]**, proponiendo una alternativa con gran potencial para el desarrollo web y la integración de sistemas con el mencionado y conocido software; aprovechando para ello, recursos externos de Software Libre.

Analizando lo expuesto anteriormente, se ha llegado al desarrollo de los siguientes puntos:

- **Apoyo a la Enseñanza**

La herramienta, permite a los profesores establecer una línea de desarrollo, basada en plantillas preconfiguradas, para la construcción del documento SRS de la asignatura por parte de los estudiantes y ofrece la posibilidad de gestionar los equipos de trabajo mediante una interfaz de usuario sencilla, sobre un sistema de información en el que los estudiantes deberán estar registrados.

- **Apoyo al Aprendizaje**

La herramienta, ofrece a los estudiantes una forma de acceder, mantener, editar, guardar y generar de forma cooperativa el documento SRS, a través de, un entorno común y guiado al que pueden acceder todos los integrantes de un equipo de trabajo; de manera que puedan ir desarrollando cada apartado del documento con relativa facilidad y organización conjunta.

- **Soluciones Web**

Otro de los objetivos de la herramienta, además de ofrecer una interfaz intuitiva y adecuada para los usuarios basada en preceptos de UX, pretende como parte de sus objetivos, exponer tecnologías **web actuales**; planteando un desarrollo centralizado en la parte front-end, que se encargará de mantener la mayor parte de la funcionalidad en cliente.

- **El mundo SAP**

No es desconocido, para quienes conocen algo de **SAP [1]**, que es una de las soluciones empresariales más utilizada y robusta del mercado. Pero de igual forma, en la práctica conlleva unos costes de licencias y personal muy elevados.

Actualmente, **SAP [1]** ofrece una solución de desarrollo de aplicaciones web altamente usables, en pleno auge, a través del framework front-end **SAP UI5 [2]** pensado para integrarse con un back-end

en máquinas **SAP**; y muchas son las empresas, que deciden incursionar en una licencia SAP para llevar **solamente algunas** partes de su gestión, pues debido a los costes, suelen dejar otras partes en sistemas más económicos o que ya han utilizado anteriormente.

Por ese motivo, este trabajo pretende exponer una solución, quizá atrevida, mostrando cómo el Framework **SAP UI5** (utilizado en este trabajo), puede integrarse con otras tecnologías back-end de **Software Libre**.

Por lo tanto, en una visión más periférica, lo que se plantea como objetivo, es ofrecer un puente de integración entre sistemas **SAP** y **otro tipo de sistemas** a través de un mismo desarrollo web front-end, que pueda atacar tanto Servicios SAP, como otros servicios Externos (en este caso servicios basados en Node.js y Mongo DB).

2.2 Motivación

En este aspecto y desde la perspectiva personal como estudiante, ha sido y es de vital importancia la realización de un trabajo que sirva para apoyar a más personas y colectivos. Es de ahí, que nace la motivación para el desarrollo de una herramienta que permite al autor, compartir conocimientos y ayudar al desarrollo, enseñanza y aprendizaje de las bases de la **Ingeniería del Software**, siendo ésta, una de las asignaturas que marcará la diferencia a la hora de incorporarse en un mercado laboral.

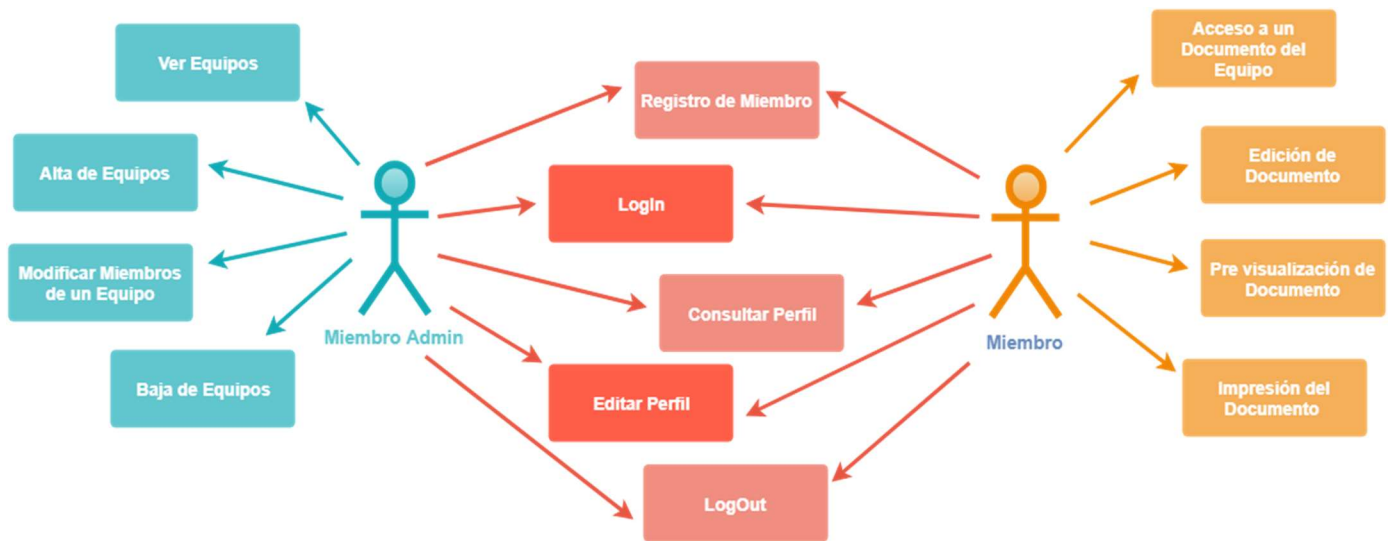
Además, otro de los motivos de la realización, y hablando desde un ámbito profesional, es el hecho de dar a conocer un trabajo desarrollado mayormente en la parte front-end, exponiendo la capacidad de tecnologías web (entre las que deseo destacar **SAPUI5**), actualmente demandadas y poco estudiadas; con el fin de compartir conocimientos que serán de utilidad a quienes puedan acceder al desarrollo de este trabajo apostando por un campo sólido e importante como es el mundo de las tecnologías **SAP**.

3. Especificación de Requisitos

3.1 Casos de Uso

Las distintas funcionalidades de la herramienta desarrollada, **IS Case Tool – Docs**, se detallan en este capítulo de manera formal, explícita y coherente de acuerdo con los fundamentos de la Especificación de Requisitos.

Para ello, se ofrecen los siguientes casos de uso [Figura 1.], descritos mediante un diagrama UML:



[Figura 1: Diagrama de Casos de Uso]

3.2 Características del Usuario

El Sistema denomina a los usuarios “**Miembros**” y como podemos apreciar en [Figura 1.] contempla dos tipos:

- **Miembro Administrador**, que tendrá disponibles, además de las funcionalidades generales que tratan sus propios datos; funcionalidades de gestión de Equipos de trabajo de la asignatura. En contexto, este usuario haría el Rol de Profesor.
- **Miembro**, que tendrá disponibles, además de las funcionalidades generales que tratan sus propios datos; funcionalidades dedicadas a la construcción, edición, almacenamiento e impresión del Documento SRS.

3.3 Requisitos Funcionales

A continuación, realizamos un detalle de los casos de uso descritos en [Figura 1.] donde se aprecia la existencia de tres bloques diferenciados:

- **Funciones Comunes a Todos los Miembros:**

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU1 |
| Nombre | Registro de Miembro |
| Actores | Miembro ó Miembro Administrador |
| Precondiciones | Configuración de la Herramienta, mediante un fichero (config. json), para determinar el Rol (Admin/ No Admin) de Registro mediante un flag. |
| Post Condiciones | El Usuario recibe un mensaje de confirmación de alta como Miembro a través de la UI y puede proceder a autenticarse. |
| Descripción | Cada usuario debe realizar un registro para darse de alta como Miembro del Sistema, a través de un formulario, facilitando los datos requeridos. El proceso se realiza desde la vista principal de la UI en IS Case Tool - Docs . |
| Flujo Principal | <ol style="list-style-type: none"> 1. El Usuario accede a la vista principal del sistema utilizando un web Browser. 2. El usuario presiona el botón de “Registro” que presenta un formulario a rellenar con sus datos. 3. Tras rellenar todos sus datos el sistema lo valida y activa el botón de confirmación de Registro. 4. El usuario presiona el botón de Confirmación. 5. Los datos se almacenan en el sistema, y el Usuario queda dado de Alta. 6. Se muestra un mensaje de confirmación del Alta. |
| Flujo Alternativo | <ol style="list-style-type: none"> 1. El Usuario accede a la vista principal del sistema utilizando un web Browser. |

| | |
|--|--|
| | <ol style="list-style-type: none"> 2. El usuario presiona el botón de “Registro” que presenta un formulario a rellenar con sus datos. 3. Tras rellenar todos sus datos el sistema No valida los Datos por haber algún error en los mismos. 4. El usuario corrige los datos erróneos. 5. Tras rellenar todos sus datos el sistema lo valida y activa el botón de confirmación de Registro. 6. El usuario presiona el botón de Confirmación. 7. Los datos se almacenan en el sistema, y el usuario queda dado de Alta. 8. Se muestra un mensaje de confirmación del Alta. |
|--|--|

| | |
|-------------------------------|--|
| Identificador del Caso de Uso | CU2 |
| Nombre | LogIn |
| Actores | Miembro ó Miembro Administrador |
| Precondiciones | El usuario debe haber realizado un Registro de Miembro (CU1) correcto en el sistema. |
| Post Condiciones | El usuario, accede al catálogo de funcionalidades ofrecidas “LaunchPad”. |
| Descripción | Un usuario registrado, accede al sistema mediante un formulario de autenticación. Pudiendo así visitar la interfaz “Launchpad” que da acceso a todas las funcionalidades que se le ofrecen. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario accede a la vista principal del sistema utilizando un web Browser. 2. El usuario rellena los datos de autenticación mostrados en el formulario de LogIn. 3. El usuario presiona el Botón de LogIn. |

| | |
|-------------------|---|
| | <ol style="list-style-type: none"> Los datos introducidos son evaluados por el sistema y se validan correctamente. El sistema configura una sesión sobre el web Browser. El usuario accede al sistema, pudiendo interactuar con el catálogo de funcionalidades. |
| Flujo Alternativo | <ol style="list-style-type: none"> El usuario accede a la vista principal del sistema utilizando un web Browser. El usuario rellena los datos de autenticación mostrados en el formulario de LogIn. El usuario presiona el Botón de LogIn. Los datos introducidos son evaluados por el sistema y No se validan por ser erróneos. Se muestra un mensaje de error en la autenticación. |

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU3 |
| Nombre | LogOut |
| Actores | Miembro ó Miembro Administrador |
| Precondiciones | El usuario, ha realizado un LogIn (CU2) y accede al catálogo de funcionalidades ofrecidas "LaunchPad". |
| Post Condiciones | El listado de Equipos se muestra actualizado de acuerdo al filtro aplicado. |
| Descripción | El usuario puede cerrar su sesión y volver a la vista inicial de la aplicación desde el catálogo "LaunchPad"; borrando de forma segura así sus datos de sesión. |
| Flujo Principal | <ol style="list-style-type: none"> El usuario presiona el Botón de LogOut. |

| | |
|-------------------|---|
| | <ol style="list-style-type: none"> 2. El sistema Limpia los datos de sesión vigentes sobre el Web Browser. 3. El sistema muestra la interfaz inicial de la herramienta. (CU2) |
| Flujo Alternativo | <ol style="list-style-type: none"> 1. El usuario cierra el web Browser. 2. El sistema Limpia los datos de sesión vigentes sobre el Web Browser. 3. El sistema muestra la interfaz inicial de la herramienta. (CU2) |

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU4 |
| Nombre | Consultar Perfil |
| Actores | Miembro ó Miembro Administrador |
| Precondiciones | El usuario ha sido autenticado (CU2) y ha accedido al catálogo de Funcionalidades disponibles "LaunchPad" |
| Post Condiciones | El usuario puede ver correctamente sus datos personales y utilizar la funcionalidad ofrecida para gestionarlos. |
| Descripción | Cada usuario puede consultar sus datos personales, facilitados tras su registro, accediendo a la interfaz de "Mi Perfil" disponible en el catálogo previo "LaunchPad". |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona sobre el Tile correspondiente a "Mi Perfil" disponible en el Catálogo. 2. El Sistema recupera los datos del usuario almacenados. 3. El sistema presenta los datos del Miembro a través de la UI en la vista correspondiente. |
| Flujo Alternativo | |

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU5 |
| Nombre | Editar Perfil |
| Actores | Miembro ó Miembro Administrador |
| Precondiciones | El usuario ha sido autenticado y ha accedido a la UI de “Mi Perfil” (CU4) mediante el Tile correspondiente del catálogo “LaunchPad”. |
| Post Condiciones | Todos los datos actualizados por el usuario quedan almacenados en el sistema y se muestran correctamente. |
| Descripción | Tras acceder a su perfil el usuario es capaz de modificar los datos personales proporcionados durante su registro y almacenarlos actualizados en el sistema, a través de la funcionalidad que ofrece la UI. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona sobre cualquiera de los botones de edición de la vista “Mi Perfil” en la UI. 2. El sistema habilita los campos de datos a Modificar. 3. El usuario edita los datos. 4. El usuario confirma los cambios. 5. El sistema actualiza los datos almacenados. 6. El sistema muestra un mensaje que confirma una actualización correcta |
| Flujo Alternativo | |

- **Funciones de Miembro Administrador:**

| | |
|-------------------------------|-----|
| Identificador del Caso de Uso | CU6 |
|-------------------------------|-----|

| | |
|-------------------|---|
| | |
| Nombre | Ver Equipos |
| Actores | Miembro Administrador |
| Precondiciones | El usuario ha sido autenticado (CU2) y ha accedido al catálogo de Funcionalidades disponibles "LaunchPad". |
| Post Condiciones | El usuario es capaz de ver y utilizar las funcionalidades ofrecidas para la Gestión de Equipos de Trabajo. |
| Descripción | El usuario puede consultar los nombres y miembros de los equipos de trabajo dados de alta en el sistema, accediendo a la interfaz de "Gestionar Equipos" ofrecida en el catálogo "LaunchPad" y seleccionando un equipo de entre una lista de ellos; para a continuación ver la información detallada del equipo seleccionado. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona sobre el Tile correspondiente a "Gestionar Equipos" disponible en el Catálogo. 2. El Sistema recupera los datos de los equipos almacenados. 3. El sistema presenta los datos de los equipos dados de alta a través de la UI en la vista correspondiente. |
| Flujo Alternativo | |

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU7 |
| Nombre | Búsqueda de Equipos |
| Actores | Miembro Administrador |
| Precondiciones | El usuario ha sido autenticado y accedido a la UI de "Gestionar Equipos" (CU5) mediante el Tile correspondiente del catálogo "LaunchPad". |
| Post Condiciones | El usuario visualiza el listado de equipos filtrado por el nombre introducido. |

| | |
|-------------------|---|
| | |
| Descripción | El usuario puede buscar equipos a través de un input de búsqueda, cuya función es mostrar los equipos con el nombre o parte del nombre que ha introducido el usuario. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario escribe el nombre de un equipo o parte del mismo en el input de búsqueda. 2. El Sistema filtra el listado de equipos por nombre. 3. El sistema actualiza el listado de equipos que se muestran y los presenta al usuario. |
| Flujo Alternativo | |

| | |
|-------------------------------|--|
| Identificador del Caso de Uso | CU8 |
| Nombre | Alta de Equipo |
| Actores | Miembro Administrador |
| Precondiciones | El usuario ha sido autenticado y accedido a la UI de “Gestionar Equipos” (CU5) mediante el Tile correspondiente del catálogo “LaunchPad”. |
| Post Condiciones | Un nuevo equipo de trabajo se ha dado de alta en el sistema. |
| Descripción | El usuario da de alta un nuevo equipo de trabajo, escogiendo a los miembros que lo conforman, de entre los ya registrados, y dándole un nombre al equipo. De esta manera un equipo y sus documentos, en principio sin contenido, quedan almacenados en el sistema. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona sobre el botón de “Nuevo Equipo”. 2. El sistema recupera los datos de los miembros disponibles y muestra una lista seleccionable al usuario, así como un campo para escribir el nombre del nuevo equipo. |

| | |
|-------------------|--|
| | <ol style="list-style-type: none"> 3. El usuario escribe El nombre del equipo y selecciona uno o más miembros. 4. El botón de confirmación de Creación se activa. 5. El usuario presiona el botón de confirmación de Creación. 6. El sistema Crea un nuevo equipo, relaciona el equipo con sus miembros y da de alta todos los documentos y apartados en principio vacíos del nuevo equipo. 7. Se muestra un mensaje de confirmación, asegurando la creación correcta del equipo. |
| Flujo Alternativo | |

| | |
|-------------------------------|--|
| Identificador del Caso de Uso | CU9 |
| Nombre | Modificar Miembros de un Equipo |
| Actores | Miembros y/o Miembros Administradores |
| Precondiciones | El usuario ha sido autenticado y accedido a la UI de “Gestionar Equipos” (CU5) mediante el Tile correspondiente del catálogo “LaunchPad”. Y además ha seleccionado uno de los equipos existentes. |
| Post Condiciones | Toda la información de un equipo se muestra de manera correcta y actualizada. |
| Descripción | El usuario agrega o elimina miembro/s en un equipo en el sistema utilizando para ello, los botones ofrecidos por la UI en la vista correspondiente. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario selecciona un equipo y visualiza los detalles del mismo. 2. El usuario presiona el botón de “Agregar Miembro” 3. El sistema recupera los datos de los miembros disponibles y muestra una lista seleccionable al usuario. |

| | |
|-------------------|--|
| | <ol style="list-style-type: none"> 4. El usuario selecciona uno a más miembros para agregar al equipo. 5. El botón de confirmación de la agregación se Activa. 6. El usuario presiona el botón de Confirmación. 7. El sistema agrega y almacena la información con el/los miembros agregados. 8. El sistema muestra la información de los miembros de un equipo actualizada. |
| Flujo Alternativo | <ol style="list-style-type: none"> 1. El usuario selecciona un equipo y visualiza los detalles del mismo. 2. El usuario presiona sobre el botón “Eliminar” asociado a uno de los miembros de un equipo. 3. El sistema, actualiza la información del equipo almacenada eliminando el miembro seleccionado. 4. El sistema muestra la información de los miembros de un equipo actualizada. |

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU10 |
| Nombre | Baja de Equipo |
| Actores | Miembros y/o Miembros Administradores |
| Precondiciones | El usuario ha sido autenticado y accedido a la UI de “Gestionar Equipos” (CU6) mediante el Tile correspondiente del catálogo “LaunchPad”. |
| Post Condiciones | Toda la información sobre los equipos almacenada y mostrada se encuentra actualizada. |
| Descripción | El usuario elimina un equipo del sistema utilizando para ello, los botones ofrecidos por la UI en la vista correspondiente. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario selecciona un equipo. |

| | |
|-------------------|--|
| | <ol style="list-style-type: none"> 2. El usuario presiona el botón de “Eliminar equipo”. 3. El sistema requiere al usuario una confirmación de la Eliminación. 4. El usuario confirma la eliminación. 5. El sistema elimina todos los datos almacenados referentes al equipo, incluyendo contenidos y documentos. 6. El sistema elimina el equipo almacenado. 7. El sistema muestra un mensaje de confirmación de la eliminación correcta. 8. El sistema muestra la lista de equipos actualizada. |
| Flujo Alternativo | <ol style="list-style-type: none"> 1. El usuario selecciona un equipo. 2. El usuario presiona el botón de “Eliminar equipo”. 3. El sistema requiere al usuario una confirmación de la Eliminación. 4. El usuario cancela la eliminación. 5. El sistema muestra la lista de equipos actualizada. Sin realizar cambios. |

• **Funciones de un Miembro:**

| | |
|-------------------------------|--|
| Identificador del Caso de Uso | CU11 |
| Nombre | Acceso al Documento SRS del Equipo |
| Actores | Miembro |
| Precondiciones | El usuario ha sido autenticado (CU2) y ha accedido al catálogo de Funcionalidades disponibles “LaunchPad”. |

| | |
|-------------------|--|
| Post Condiciones | El sistema muestra el Documento y habilita al usuario para el uso de sus funcionalidades. |
| Descripción | <p>El usuario puede ver todos y cada uno de los apartados y contenidos del documento de su equipo, accediendo a la interfaz “SRS” ofrecida en el catálogo y que muestra una vista intuitiva, organizada y preparada para la edición guiada del documento por parte del usuario.</p> <p>Para ello, se muestra la lista de apartados (organizados según plantilla preconfigurada). De forma que al seleccionar cualquiera de ellos se presenta el detalle y funcionalidad de los subapartados y contenidos del apartado seleccionado.</p> |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona sobre el Tile correspondiente a “SRS” disponible en el Catálogo. 2. El Sistema recupera los datos del documento almacenados. 3. El sistema presenta el documento correspondiente al equipo del usuario que accede a través de la UI en la vista correspondiente. 4. El sistema o usuario, selecciona uno de los apartados del documento. 5. El sistema muestra en detalle los subapartados y contenidos del Apartado seleccionado y habilita sus funciones. |
| Flujo Alternativo | |

| | |
|-------------------------------|--|
| Identificador del Caso de Uso | CU12 |
| Nombre | Edición del Documento SRS |
| Actores | Miembro |
| Precondiciones | El usuario ha sido autenticado y accedido a la UI de “SRS” (CU11) mediante el Tile correspondiente del catálogo “LaunchPad”. |

| | |
|-------------------|---|
| | |
| Post Condiciones | Todo el contenido editado del documento SRS es almacenado correctamente en el sistema y se encuentra disponible en la UI. |
| Descripción | El usuario tras seleccionar uno de los apartados del documento, puede acceder a los subapartados y contenidos del mismo (en inicio vacíos) con el fin de Editarlos. Mediante el botón de “Guías de Uso”, obtiene ayuda sobre como rellenar cada contenido. Tras modificar cualquiera de los contenidos de cualquier Apartado el sistema activa la función de “Auto Guardado” dejando toda la información almacenada y disponible. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario selecciona un apartado de la lista presentada (Índice del Documento preconfigurado). 2. El sistema recupera la información y muestra los subapartados y contenidos editables. 3. El usuario lee la ayuda ofrecida tras seleccionar el botón “Guías de Uso” 4. El usuario selecciona alguno de los campos editables. 5. El usuario edita el contenido. 6. El usuario abandona el foco del campo editado. 7. El sistema activa el “Auto Guardado” y almacena la información del contenido editado. 8. El sistema muestra un mensaje confirmando haber salvado los cambios. |
| Flujo Alternativo | <ol style="list-style-type: none"> 1. El usuario selecciona un apartado de la lista presentada (Índice del Documento preconfigurado). 2. El sistema recupera la información y muestra los subapartados y contenidos editables. 3. El usuario selecciona alguno de los campos editables. 4. El usuario edita el contenido. 5. El usuario abandona el foco del campo editado. |

| | |
|--|--|
| | <p>6. El sistema activa el “Auto Guardado” y almacena la información del contenido editado.</p> <p>7. El sistema muestra un mensaje confirmando haber salvado los cambios.</p> |
|--|--|

| | |
|-------------------------------|--|
| Identificador del Caso de Uso | CU13 |
| Nombre | Previsualización del Documento SRS |
| Actores | Miembro |
| Precondiciones | El usuario ha sido autenticado y accedido a la UI de “SRS” (CU11) mediante el Tile correspondiente del catálogo “LaunchPad”. |
| Post Condiciones | Se muestra una previsualización del documento final y se habilita la impresión. |
| Descripción | El usuario puede acceder a la previsualización del SRS final presionando el botón correspondiente; se mostrará una vista con todo el contenido Actual del documento y con un formato adecuado para la impresión. De forma que el usuario pueda comprobar y decidir si todo está listo para generar el PDF final. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona el botón “Previsualizar pdf” (disponible bajo la lista de Apartados). 2. El sistema recupera construye una visualización adecuada con los datos almacenados. 3. El sistema presenta una visualización del documento final. 4. El sistema activa la funcionalidad de impresión. |
| Flujo Alternativo | |

| | |
|-------------------------------|---|
| Identificador del Caso de Uso | CU14 |
| Nombre | Impresión del Documento SRS |
| Actores | Miembro |
| Precondiciones | El usuario autenticado, ha accedido a la UI de "Previsualización pdf" (CU11). |
| Post Condiciones | El usuario obtiene el documento impreso en formato PDF. |
| Descripción | El usuario puede imprimir el documento presionando el botón correspondiente, de forma que se genera un documento PDF sobre la interfaz preparada para ello, del propio web Browser. |
| Flujo Principal | <ol style="list-style-type: none"> 1. El usuario presiona el botón "Imprimir" desde la previsualización. 2. El sistema activa los parámetros de impresión. 3. El sistema utiliza el web Browser, para generar un documento pdf. 4. El Web Browser presenta el documento descargable en formato PDF. |
| Flujo Alternativo | |

4. Tecnologías Aplicadas

Una de las decisiones más importantes y cruciales a la hora de realizar el desarrollo, ha sido escoger las tecnologías aplicadas. A continuación, se presentan organizadas:

- **Front-End:** dado que el proyecto se centra un 90% en el desarrollo front-end la decisión más importante ha sido seleccionar y exponer una tecnología adecuada, en este caso **FIORI - SAP UI5**.

SAP UI5 [2], es un Framework desarrollado por **SAP** que sigue una metodología específica basada en Preceptos de UX (Fiori [3]).

Basado en Estándares de HTML 5 [4], JavaScript (ECMA5) [5], oData [6] y CSS3 [7] recoge un conjunto de librerías que permiten desarrollar aplicaciones web empresariales.

Actualmente, es uno de los Frameworks Front-End más demandados y poco conocidos sobre el mundo SAP y por tanto en general, sobre el mundo de la programación Web.

Aunque, SAP UI5 [2] está pensado para un entorno SAP, ofreciendo facilidades para trabajar contra un back-end desarrollado con el conocido software. No es menos cierto que, plantea una comunicación y manejo de datos a través del protocolo **oData [6]**(sw-libre), de manera que se puede atacar a cualquier tipo de back-end capaz de manejar dicho protocolo y es exactamente lo que se plantea en este trabajo.

Finalmente, en la práctica y con motivo de ofrecer un desarrollo en Software Libre. Se ha decidido escoger la versión libre del framework, **Open UI5**.

La diferencia principal entre los dos Frameworks es la Licencia que llevan y que **SAP UI5** cuenta con algunas librerías extendidas que **Open UI5** no tiene. Sin embargo, el conjunto que abarca **Open UI5** es de sobra suficiente para el desarrollo de una aplicación profesional.

- **Exposición de datos y Comunicaciones:** en este sentido el desarrollo con SAPUI5 maneja un modelado de datos que requiere el uso del protocolo oData [6].

oData [6], es un protocolo Microsoft (libre) que permite la exposición y consumo de datos (relacionales y no relacionales), como Servicios a través de **URI's** ofreciendo una manera uniforme de representar datos estructurados en formato **JSON [8]**; así como, operaciones simples para petición, filtrado, orden, paginación, modificación, creación y borrado de datos basándose en los verbos http GET, POST, PUT y DELETE [9].

- **Back-end:** en este aspecto y dado que la herramienta necesita manejo de oData, se ha decidido seleccionar una librería npm (Node.js) [10] llamada **"n-odata-server"**.

n-odata-server [11], es una componente como librería, que gestiona la exposición de datos y operaciones CRUD sobre diversos tipos de BD; mediante un puerto escucha y utilizando el Protocolo oData.

Hay que mencionar, además, que "n-odata-server" se monta sobre el conocido framework **"API RestFull LoopBack" [12]**. De manera que, mediante una configuración relativamente simple, referente a Modelo de datos y conexiones, nos ofrece un servicio API RestFull que trabaja con oData y que no requiere más implementación.

- **Almacenamiento:** Para almacenar y gestionar todos los datos se ha escogido la Base de Datos no relacional MongoDB [13].

Mongo DB, es una base de datos no relacional de código abierto, que almacena información en documentos con formato JSON.

- **Servidor-web:** aunque “n-odata-server” ofrece una API RestFull expuesta, que sería suficiente para conectarse desde un PC que tenga la **implementación front-end**; se ha decidido establecer un entorno más real, colocando ésta última sobre un servidor **Apache** [14] que arranca a través de la conocida distribución Apache **XAMPP** [15] de manera sencilla y ágil. Esto permite tener toda la implementación en servidor y ofrecer al usuario una herramienta web a la que puede acceder simplemente utilizando una URL en su navegador.
- **Entorno de desarrollo y Gestión de Versiones:** en este aspecto y dado que es un trabajo individual, se ha seleccionado como entorno de desarrollo la herramienta **SAP WEB IDE** [16] ofrecida como servicio por **SAP Cloud Platform (trial Version)** [17] y que cuenta con integración **GitHub** [18], para el control de versiones.

SAP WEB IDE [16], es un entorno de desarrollo web, especializado en proyectos **SAP UI5**[2] que ofrece una serie de funcionalidades adecuadas de cara a la programación, configuración, estructura y volcado de proyectos de este tipo.

Se puede acceder a él (gratuitamente), como servicio en cloud a través de la plataforma **SAP Cloud Platform (trial Version)** [17].

SAP Cloud Platform (trial Version) [17], explican en la web, “Es una plataforma abierta como servicio en cloud, que ofrece capacidades en memoria, servicios básicos de plataforma y microservicios únicos para crear y ampliar aplicaciones inteligentes y móviles.

La plataforma está diseñada para acelerar la transformación digital ayudando a desarrollar de forma rápida, sencilla y económica, aplicaciones que se adecuan a nuestro interés, sin invertir en infraestructura local.

Basado en estándares abiertos, SAP Cloud Platform ofrece total flexibilidad y control sobre Frameworks y aplicaciones”.

En resumen, a través de esta plataforma se puede acceder a los servicios necesarios para empezar el desarrollo; en concreto se adquiere acceso al entorno de desarrollo **SAP WEB IDE** [16], y a la configuración de las conexiones que permitirán la comunicación con el Back-end API RestFull para poder arrancar nuestras aplicaciones en desarrollo.

Es importante conocer que la plataforma ofrece versiones libres (TRIAL) y de pago; como es de esperar la versión de pago ofrece infinidad de servicios, utilidades y prestaciones. Sin Embargo y para lo que a este trabajo interesa, se va a utilizar la versión TRIAL que ofrece más que suficiente; incluyendo integración con una cuenta **GitHub** [18] para el control de versiones.

5. Arquitectura y Modelo de Datos

5.1 Arquitectura de la herramienta

IS Case Tool – Docs, plantea una arquitectura por capas diferenciadas. En primera instancia la capa más externa, tiene una arquitectura **Cliente-Servidor [20]** escalable, es decir, en otra perspectiva el Servidor se convierte en Cliente de una API REST.

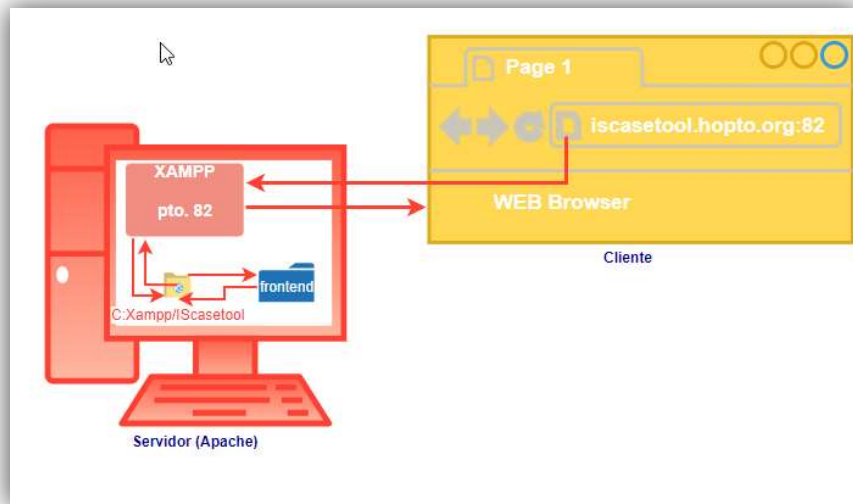
En general, la siguiente figura muestra la arquitectura completa:



[Figura 2. Arquitectura Completa del Proyecto]

A continuación, se ven estos aspectos con mayor detalle:

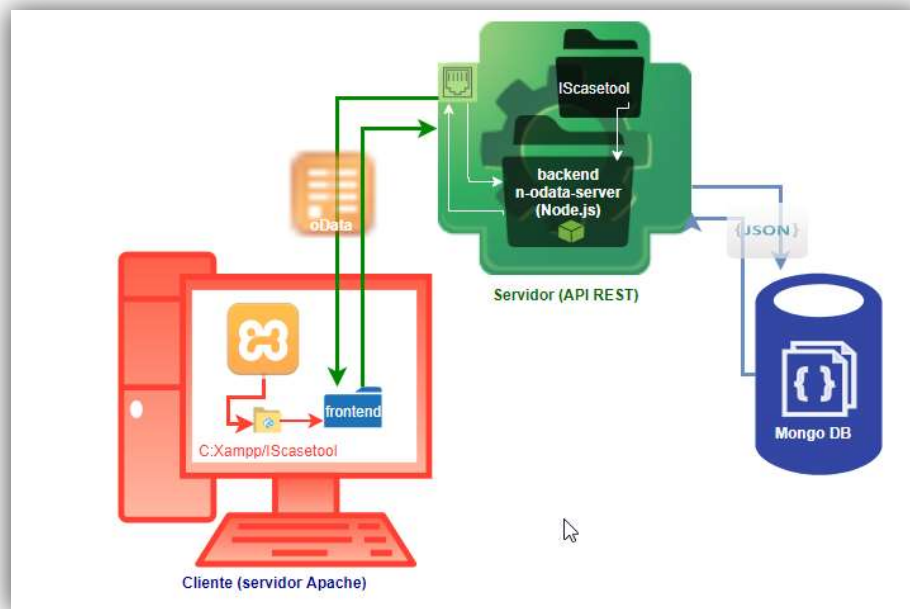
- Arquitectura en la Capa Más Externa (Cliente - Servidor):**
 En la capa más externa, encontramos una arquitectura Cliente-Servidor, en la que el Cliente viene a ser el **web Browser** del usuario final, mientras que el servidor(escalable) es la **máquina en la que se ha levantado un Apache**, encargado de servir las páginas web.



[Figura 3. Arquitectura Capa Externa]

- **Arquitectura en la Capa Secundaria (Cliente - Servidor):**

En esta capa, en cambio, encontramos una arquitectura Cliente-Servidor, en la que el Cliente viene a ser la máquina que sirve las páginas front-end SAPUI5, mientras que el servidor es una máquina en la que se levanta la API REST (n-odata-server) conectada a la anterior, como servicio a través de un puerto y que a su vez se conecta a una base de Datos Mongo DB. Dicha configuración se hace a través de un fichero simple en el back-end.

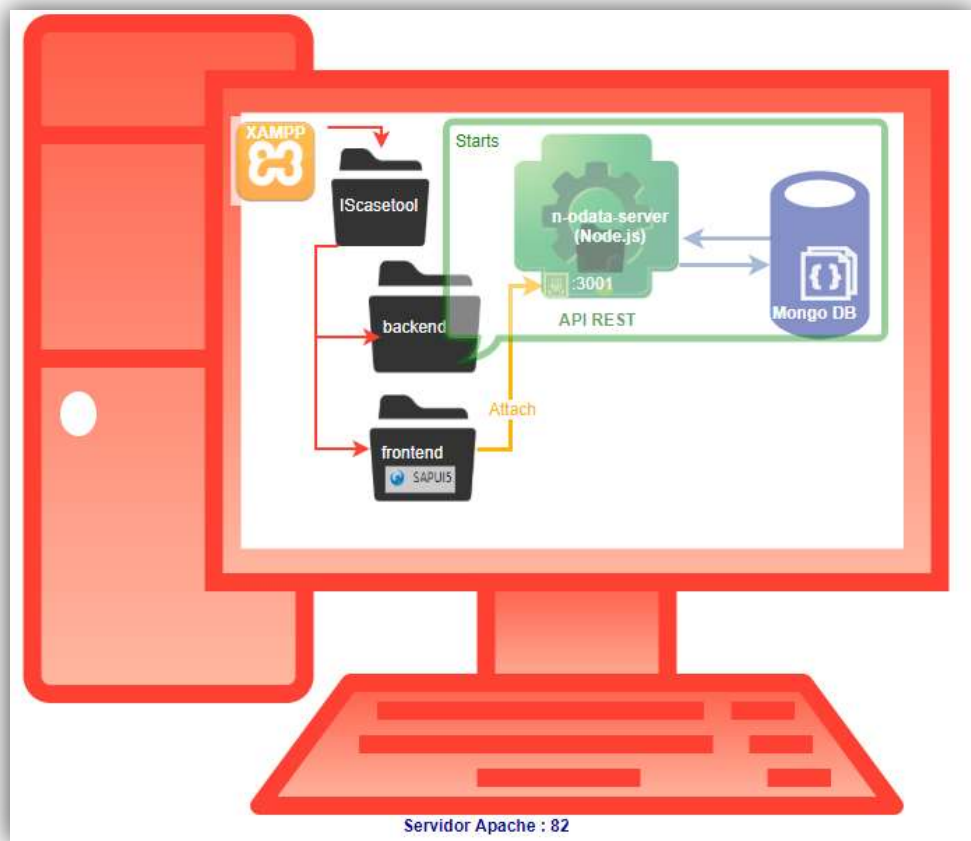


[Figura 4. Arquitectura Capa Secundaria]

- **La Arquitectura sobre el trabajo presentado**

Actualmente, con motivo de la presentación de este trabajo y por tiempo limitado, se ha montado toda la arquitectura **en una sola** máquina personal utilizando:

- Una estructura de ficheros conjunta y ubicada en el servidor Apache como directorio principal de XAMPP
- Para la Capa Externa, un DynDNS [19] a través del puerto 82 "**http://iscasetool.hopto.org:82**".
Aunque en la práctica, se puede construir esta parte sobre **cualquier servidor Apache** que sirva las páginas de la parte Front-end (SAP UI5).
- Para la Capa secundaria, se configura la conexión a la base de datos Mongo DB local, mediante un fichero de configuración en el back-end. Mientras que la API REST (n-odata-server) se levanta sobre el puerto 3001; que es el puerto al que **atacará** el front-end.



[Figura 5. Arquitectura Presentada para Demo]

5.2 Arquitectura de la Implementación

Para realizar la implementación de la herramienta, se sigue una arquitectura MVC (modelo - vista - controlador) separando la interfaz de usuario y la lógica de implementación en tres partes

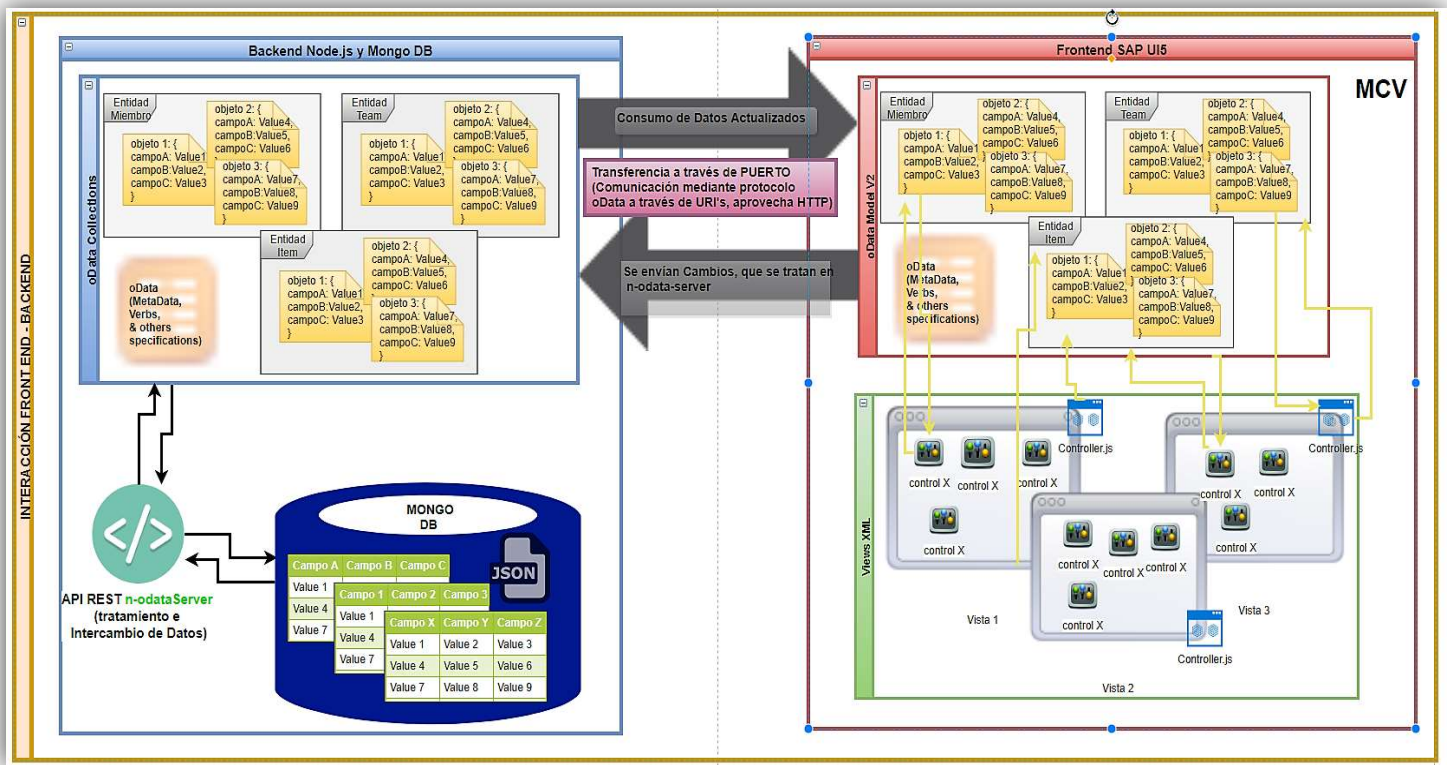
El Modelo, se encarga de manejar la representación de los datos y los mecanismos de persistencia.

La Vista, se encarga de componer y presentar la interfaz de Usuario consumiendo los datos entregados por el Modelo.

El Controlador, trabaja como punto intermedio entre la Vista y el Modelo, gestionando las transformaciones, manejo de datos e interacciones de usuario para establecer una vía de flujo entre las dos partes anteriores.

Si bien es cierto, que el modelo final persiste en servidor sobre una base de datos, decimos que el proyecto es un 90% Front-end debido a que **SAP UI5**, trabaja con su propio **MVC**, componiendo su propio modelo a partir de una imagen de los datos persistentes. Todos los flujos de datos se trabajan sobre el **Modelo Front-end (oData Model)** y cuando todas las operaciones o cambios han sido hechos sobre él, finalmente se envían a servidor donde solamente se realiza la lectura o persistencia de los cambios.

El siguiente diagrama muestra un ejemplo del funcionamiento comentado:



[Figura 6. MVC en Front-end e Interacción con Back-end]

5.3 Modelo de Datos

IS Case Tool – Docs, plantea un modelo de datos basado en el protocolo oData y que se traduce finalmente, comúnmente se conoce, que se trata de bases de datos no relacionales, sin embargo, se pueden ir construyendo relaciones mediante foreign Keys de acuerdo con el modelado que se precisa.

Dicho modelado se realiza sobre el servicio oData (n-odata-server), definiendo las distintas Entidades y relaciones que representan la información de Equipos, Miembros, Documentos, Items y Contenidos. Una vez hecho esto, es el propio servicio quien da de alta las Entidades sobre MongoDB (Colecciones) y nos proporcionará la estructura a través de una simple llamada URL. En el caso de la demo preparada para este trabajo, por ejemplo:

[http://iscasetool.hopto.org:3001/odata/\\$metadata](http://iscasetool.hopto.org:3001/odata/$metadata)

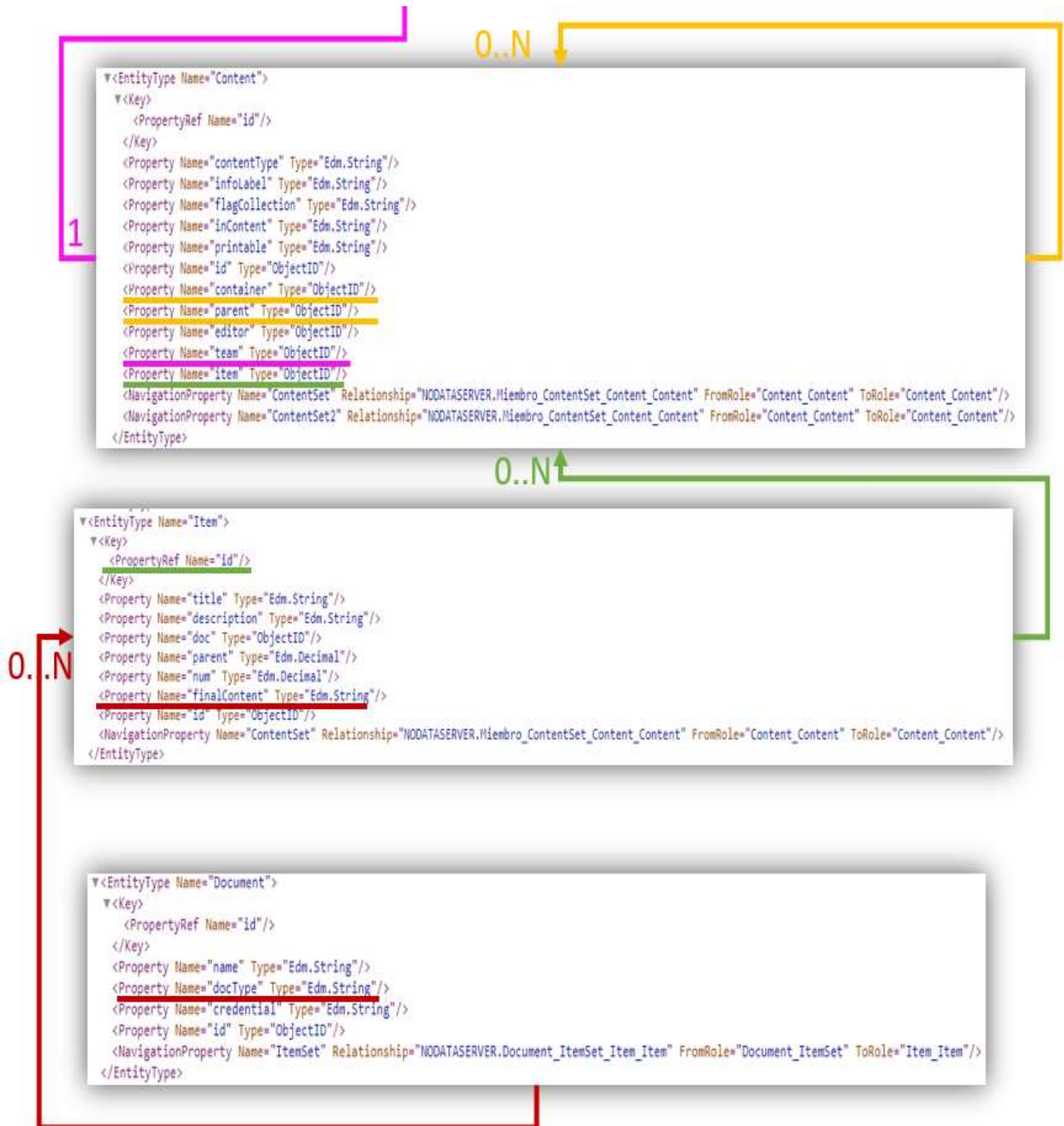
Para representar la información se decide crear las siguientes Entidades y relaciones:

```
<EntityType Name="Miembro">
  <Key>
    <PropertyRef Name="id"/>
  </Key>
  <Property Name="mail" Type="Edm.String"/>
  <Property Name="pass" Type="Edm.String"/>
  <Property Name="name" Type="Edm.String"/>
  <Property Name="lastName" Type="Edm.String"/>
  <Property Name="picture" Type="Edm.String"/>
  <Property Name="admin" Type="Edm.String"/>
  <Property Name="director" Type="Edm.String"/>
  <Property Name="id" Type="ObjectID"/>
  <Property Name="team" Type="ObjectID"/>
  <NavigationProperty Name="ContentSet" Relationship="NODATASERVER.Miembro_ContentSet_Content_Content" FromRole="Miembro_ContentSet" ToRole="Content_Content"/>
</EntityType>
```

0..N

```
<EntityType Name="Team">
  <Key>
    <PropertyRef Name="id"/>
  </Key>
  <Property Name="name" Type="Edm.String"/>
  <Property Name="director" Type="Edm.String"/>
  <Property Name="id" Type="ObjectID"/>
  <NavigationProperty Name="MiembroSet" Relationship="NODATASERVER.Team_MiembroSet_Miembro_Miembro" FromRole="Team_MiembroSet" ToRole="Miembro_Miembro"/>
  <NavigationProperty Name="ContentSet" Relationship="NODATASERVER.Miembro_ContentSet_Content_Content" FromRole="Content_Content" ToRole="Content_Content"/>
</EntityType>
```

1



[Figura 7. Modelo de Datos y Relaciones]

- **Entidad Team**, representa la información necesaria para definir un Equipo de Trabajo y sus atributos son:
 - id: identificador único de un equipo (Clave Primaria – Mongo Id).
 - name: nombre del Equipo de Trabajo (String).
 - director: identificador del Miembro Director del Proyecto de IS. (Mongo Id).
- **Entidad Miembro**, representa la información necesaria para definir un Miembro, es decir un usuario de la herramienta, que puede o no pertenecer algún equipo y sus atributos son:
 - id: identificador único de un equipo (Clave Primaria – Mongo Id).
 - mail: correo electrónico del usuario (String).
 - pass: clave de acceso a la herramienta, definido por el usuario (String Codificado).
 - name: nombre del Usuario (String).
 - lastName: Apellido/s del Usuario (String).
 - picture: URL o fichero base 64, de la imagen que representa el avatar o Foto del usuario (String).
 - admin: flag que especifica si un usuario tiene Rol de Administrador o no (String).
 - team: id del Equipo de trabajo (Team), que se refiere al equipo de trabajo al que pertenece un usuario (String).
 - director: flag que especifica si un usuario es director de su Equipo o No (String)
- **Entidad Document**, representa la información necesaria para definir un Documento. Desde el punto de vista de la herramienta, un **Documento** representará uno de los temas que un usuario puede tratar desde la UI en forma de **mini-Aplicaciones** partiendo de un catálogo.

Actualmente, pueden tratarse: SRS, Plan de Proyecto (no Activo), Memoria (No Activo), Mi Perfil y Gestionar Equipos (en caso de ser Admin).

Sus atributos son:

- id: identificador único de un equipo (Clave Primaria – Mongo Id).
- name: nombre del Documento (String).
- docType: tipo de Documento utilizado para diferenciar la navegación desde el catálogo de **mini-Aplicaciones** (SRS, PLAN, MEM, PROFILE, TEAMS). (String).
- credential: representa una credencial de acceso basada en el Rol que un usuario tenga en el Sistema diferenciando a todos los usuarios, solo Miembros no administradores, solo

Miembros administradores ó solo Directores de Equipo. Permite presentar un catálogo de **mini-Aplicaciones** u otro de acuerdo al tipo de Usuario (String).

- **Entidad Item**, representa la información necesaria para definir cada uno de los **Apartados** de un documento, que el usuario deberá rellenar; es decir, en conjunto los **Items** representan el índice de un documento.

Dado que un documento específico, siempre tendrá los mismos apartados, lo que se hace es definir estos datos a partir de una plantilla preconfigurada que puede ser cambiada fácilmente retocando un fichero JSON en el proyecto.

Sus atributos son:

- id: identificador único de un Item (Clave Primaria – Mongo Id).
- title: título del Item (String).
- description: descripción del Item (String).
- doc: tipo de documento al que pertenece el ítem (se refiere a un docType de la entidad Document). (String)
- parent: número del Item padre; cuando un Item (Apartado de un Documento) contiene subapartados, éstos están contenidos bajo su Item Padre. (String – se refiere al num del Item Padre)
- num: número de Item, representa el numero ordinal de un apartado sobre el documento (String)
- finalContent: especifica si un subapartado tiene más niveles de subapartados dentro de un documento. En tal caso todos los subniveles pasarán a ser **Contenidos** (Content) y este atributo permite hacer una diferenciación en la UI. (String – con valores “final”, “middle”, “last”).

- **Entidad Content**, representa la información necesaria para definir alguno de los contenidos de un Apartado (Item) sobre un documento.

Con el fin de facilitar la construcción de la UI, cada **Content**, representa un objeto Gráfico (a mostrar por pantalla) y que puede o nó tener una cadena escrita, que irá plasmada en un documento Real y será rellenada por un usuario.

Sus atributos son:

- id: identificador único de un Content (Clave Primaria – Mongo Id).
- contentType: representa el tipo de objeto Gráfico que será construido sobre la UI. En base a este valor, el Front-end decide cómo se pintará el Contenido (String).
- infoLabel: etiqueta informativa del “Content”, se refiere a una etiqueta con información predefinida, no por el Usuario, a mostrar en un objeto gráfico (String).

- inContent: contenido Real de cada elemento gráfico. Generalmente es sobre este campo, sobre el cual un usuario trabaja actualizando su valor. Se refiere en General al contenido que mostrará cada objeto Gráfico de la UI (String).
- flagCollection: especifica si el campo “inContent” ha sido almacenado como un conjunto de valores que se deben separar mediante su procesamiento (String).
- printable: representa un flag que especifica si un “inContent” debe ser presentado en la impresión final del Documento (String).
- container: representa el identificador del “Content” que hace de contenedor gráfico de un contenido (Mongo Id – se refiere al id de otros Contenidos).
- parent: representa el identificador del “Content” que actúa como Subapartado de tercer Nivel en un documento, en el caso de existir. Por tanto, no se trata de un contenedor sino de una construcción gráfica totalmente diferente, a nivel de un subapartado (Mongo Id – se refiere al id de otros Contenidos).
- editor: representa a un Miembro del equipo específico o a varios, quien/es podrán editar el contenido (Mongo Id ó String).
- team: representa el equipo al que pertenece el “Content” tratado (Mongo Id – se refiere al identificador de una entidad “Team”).
- item: representa el Item (apartado) al que pertenece el “Content” tratado (Mongo Id – se refiere al identificador de una entidad “Item”).

6. Diseño e Implementación

6.1 Principios de diseño

IS Case Tool – Docs, mayormente desarrollada con SAP UI5 se basa en los principios de diseño que dicta el método **SAP Fiori [21]**, adquirido gracias a la experiencia laboral del autor y aplicado sobre este trabajo.

Aunque existe mucha documentación sobre el diseño **SAP Fiori [21]**; no siendo el objetivo fundamental de este trabajo el contar toda su teoría, procedemos a continuación a explicar la base en la que se fundamenta ofreciendo una visión global y resumida.

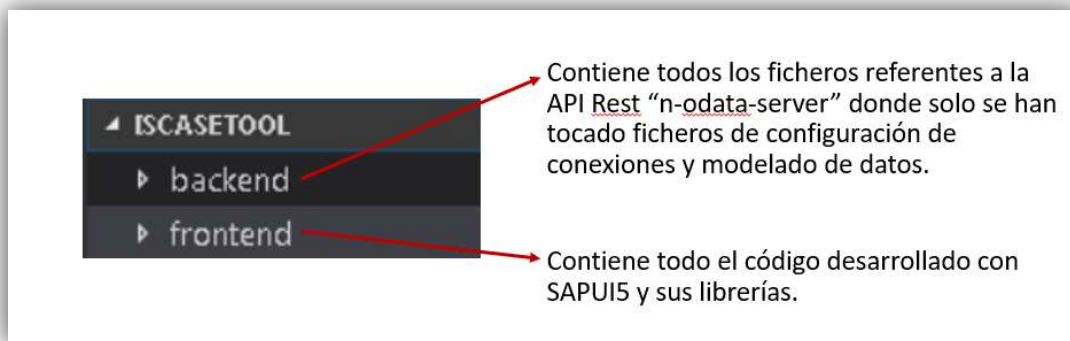
SAP Fiori [21], ofrece principios de diseño para una UX (experiencia de Usuario) moderna destinada al usuario final destacando líneas de negocio, tareas y dispositivos con un fuerte enfoque en la facilidad de uso que es intuitiva y simple; se encuentra fuertemente ligada a las librerías SAP UI5 y se basa en los siguientes principios básicos:

- **Role-Based**: especifica que los accesos se hacen desde un punto común (FIORI Launchpad – En el caso de este trabajo, el Catálogo de mini-Aplicaciones), basándose en roles y tipos de usuario multifacéticos, reflejando de esta manera la información correcta en el momento correcto formando una sólida línea de trabajo.
- **Delightful**: especifica el hecho de ofrecer componentes visuales modernos y simples enriqueciendo la experiencia de trabajo al usuario y permitiéndole simplemente realizarlo (mediante librerías SAP UI5).
- **Coherent**: especifica que sea cual sea el conjunto de áreas a tratar durante el uso de una herramienta Fiori, se debe mantener un lenguaje visual y de diseño consistente.
- **Simple**: especifica que la UI debe ayudar al usuario de forma fácil e intuitiva a concentrarse en las tareas realmente importantes que debe realizar durante su uso.
- **Adaptive**: especifica que el diseño debe permitir al usuario trabajar cómo y dónde desee, independientemente del dispositivo que utilice; proporcionando toda la información de forma eficaz e instantánea.

6.2 Interfaz de Usuario e Implementación

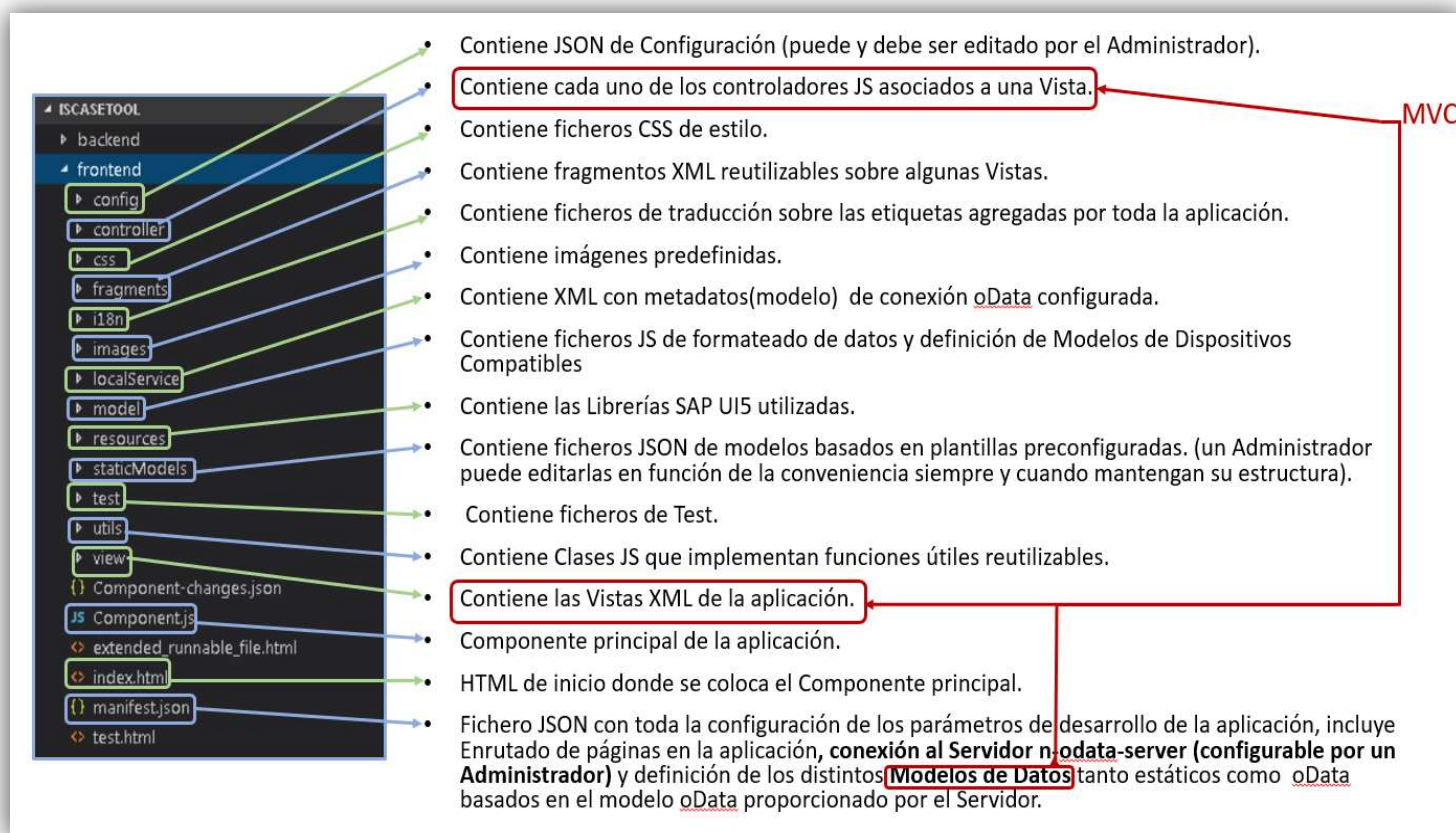
6.2.1 Visión Estructural del Código Implementado

IS Case Tool – Docs, como proyecto, se compone de dos partes esenciales, back-end y front-end que se entregan sobre un mismo directorio raíz, tal y como indica la figura:



[Figura 8. Estructura de Directorios de Implementación]

Como se puede ver, dado que el back-end solo trata configuraciones, podemos afirmar que el 90 % de la implementación desarrollada se hace en front-end y estructuralmente se compone de la siguiente manera:



[Figura 9. Estructura de Implementación Front-End]

6.2.2 Una Herramienta con Información Autogenerada

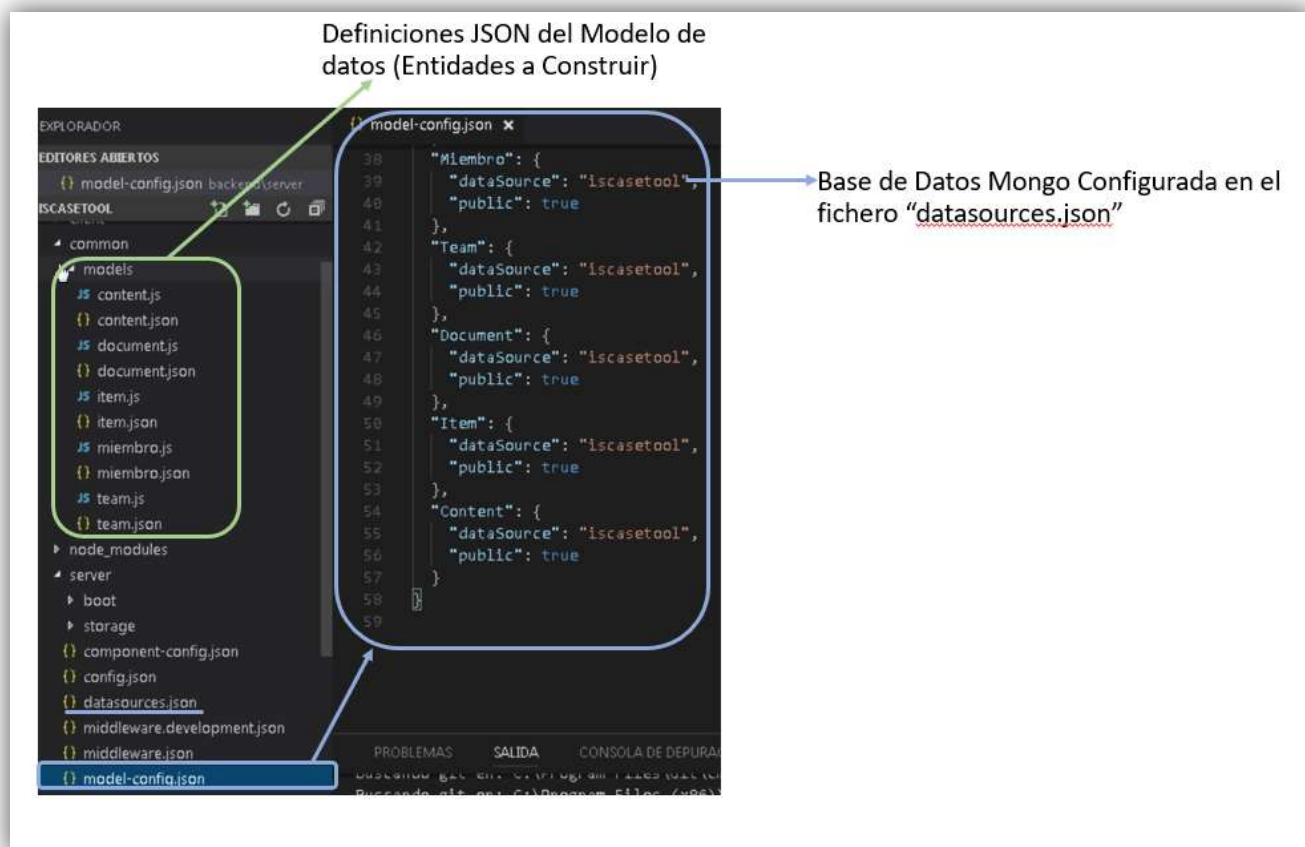
IS Case Tool – Docs, establece un modelo de datos cuya información es en su mayoría estática, exceptuando el contenido almacenado por el usuario.

Dicha información incluye Tipos de Documentos y/o Mini - Aplicaciones (Entidades Document), Índice de Apartados de un documento específico (Entidades Item) y Estructura gráfica de Contenidos sin rellenar (Entidades Content).

Teniendo en cuenta este patrón de comportamiento se ha planteado una generación automática de toda la información estática basada en plantillas JSON que se establecen en dos partes:

- **Primera parte en back-end (Creación de Colecciones en la BD)**

Mediante n-odata-server (API REST) se determina el modelo de datos que genera automáticamente las colecciones **vacías sobre la Base de Datos Mongo** a la que se conecta. Para ello sólo hace falta determinar el modelo en los ficheros de configuración "**model-config.json**" y los distintos ficheros JSON de definición de entidades bajo el directorio "**models**".



[Figura 10. Configuración de Modelos Back-end]

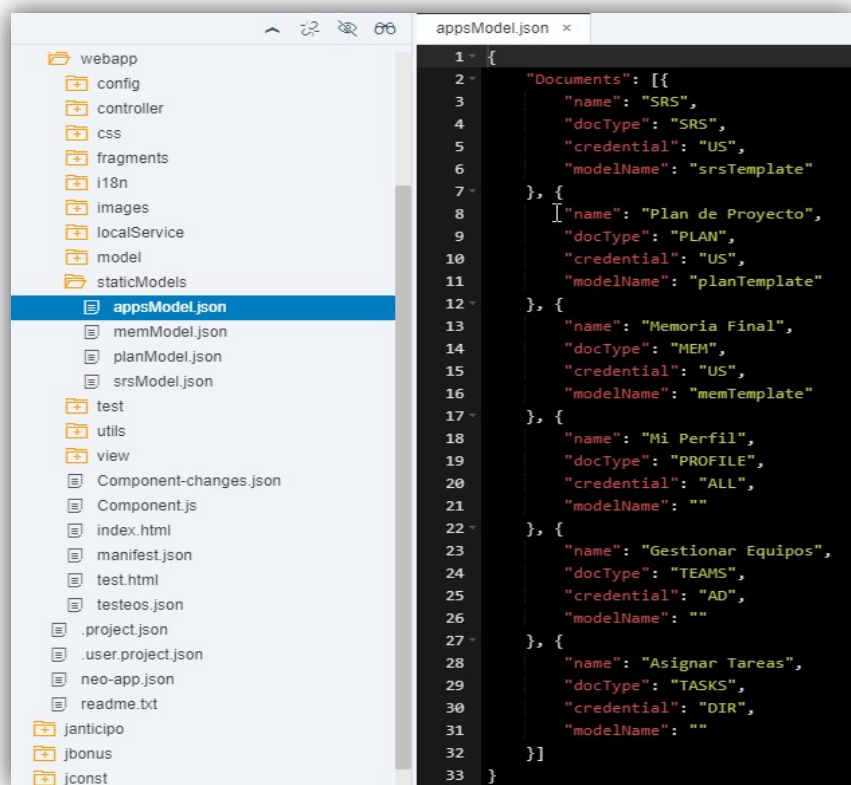


[Figura 11. Ejemplo de Definición de Modelos Back-end]

○ Segunda Parte en front-end (Rellenado de Colecciones con Datos Estáticos)

Se encarga de rellenar la base de datos Mongo con todos los **contenidos estáticos** necesarios a través de modelos JSON (no públicos) basados en **plantillas**. Para ello, se implementa en JS el rellenado contra el servidor en dos fases:

La Fase 1, se produce al arrancar la aplicación, comprobando la existencia de los datos; que en caso de no existir se dan de alta. Se Crean, los Documentos y/o Mini - Aplicaciones (Entidades Document a partir del fichero **"appsModel.json"**) y el Índice de Apartados de un documento específico (Entidades Item a partir de los ficheros **"memModel.json"**, **"planModel.json"** y **"srsModel.json"**) comunes a cualquier equipo u usuario.



[Figura 12. Plantilla para Alta de Entidades "Document" Front-end]

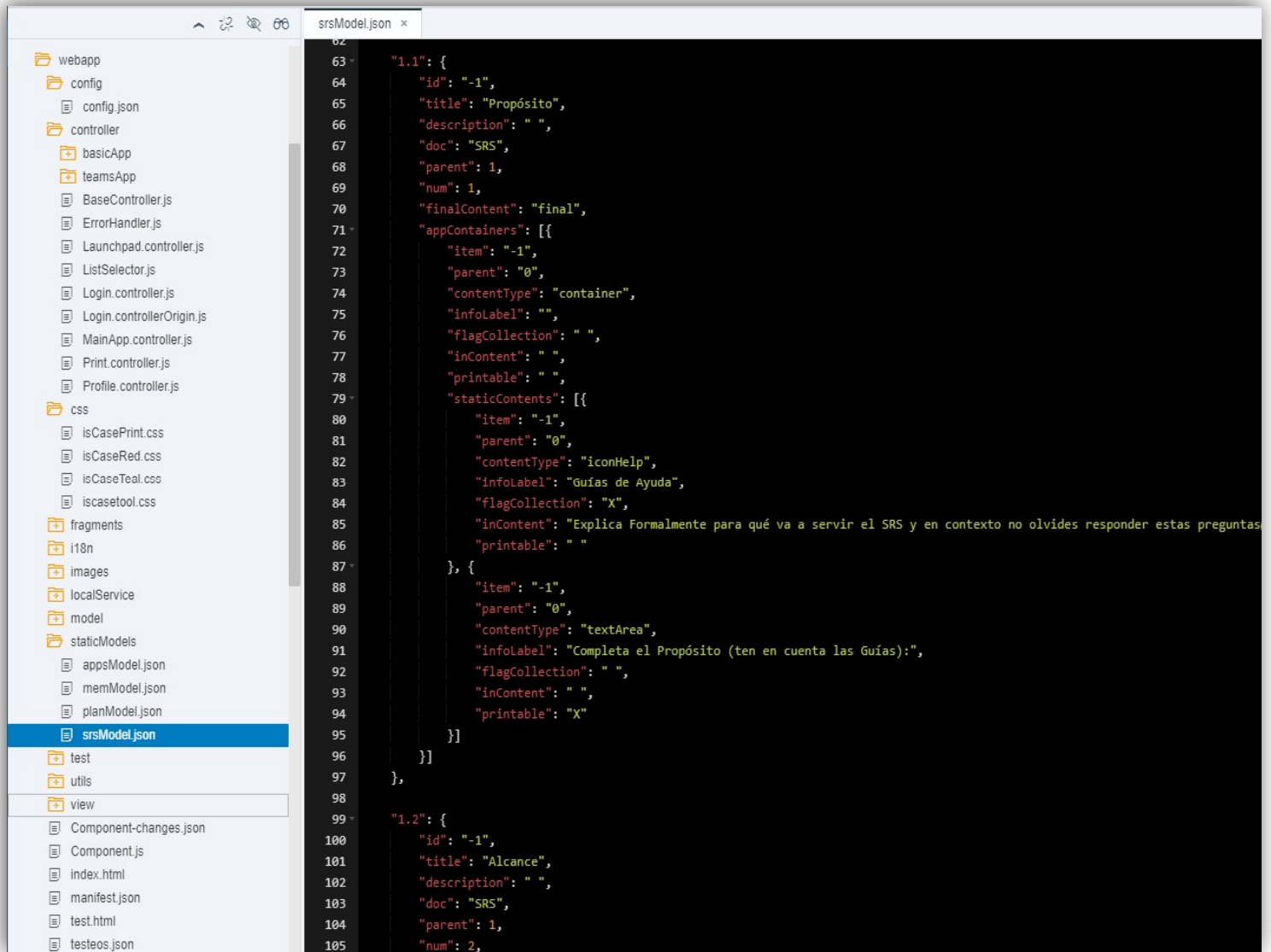
```

Component.js x
173  /*Crea todas las apps de Documentos del modelo de forma recursiva sobre la BD*/
174  _recursiveCreateDocument: function(oSelectedCollection, oIndex, oModel, oController) {
175      var oTableLimit = oSelectedCollection.length - 1;
176      var oActualData = oSelectedCollection[oIndex];
177      var oData = {
178          "name": oActualData.name,
179          "docType": oActualData.docType,
180          "credential": oActualData.credential
181      };
182      oModel.create("/DocumentSet", oData, {
183          success: function(oDataRes) {
184              if (oIndex < oTableLimit) {
185                  oController._recursiveCreateDocument(oSelectedCollection, oIndex + 1, oModel, oController);
186              } else { // una vez dadas de alta todas las apps hay que dar de alta los Items de cada App de Documento
187                  var oFinalDocCollection = null;
188                  var oModelApps = oController.getModel("Apps");
189                  var oCollectionToUpload = oModelApps.getData().Documents;
190                  if (oCollectionToUpload.length > 0) {
191                      oFinalDocCollection = oCollectionToUpload;
192                  }
193                  if (oFinalDocCollection) {
194                      for (var i = 0; i < oFinalDocCollection.length; i++) {
195                          var oCurrentApp = oFinalDocCollection[i];
196                          if (oCurrentApp.credential === "US" && oCurrentApp.modelName && oCurrentApp.docType) {
197                              var appFilter = new sap.ui.model.Filter({
198                                  path: "doc",
199                                  operator: sap.ui.model.FilterOperator.EQ,
200                                  value1: oCurrentApp.docType
201                              });
202                              oController._loadStaticModel_AppsContent(appFilter, oCurrentApp.modelName);
203                          }
204                      }
205                  } else {
206                      //Incluir mensaje de Error en alta y consulta de Apps
207                      return;
208                  }
209                  return;
210              }
211          },
212          error: function(oError) {
213
214          }
215      });
216  },

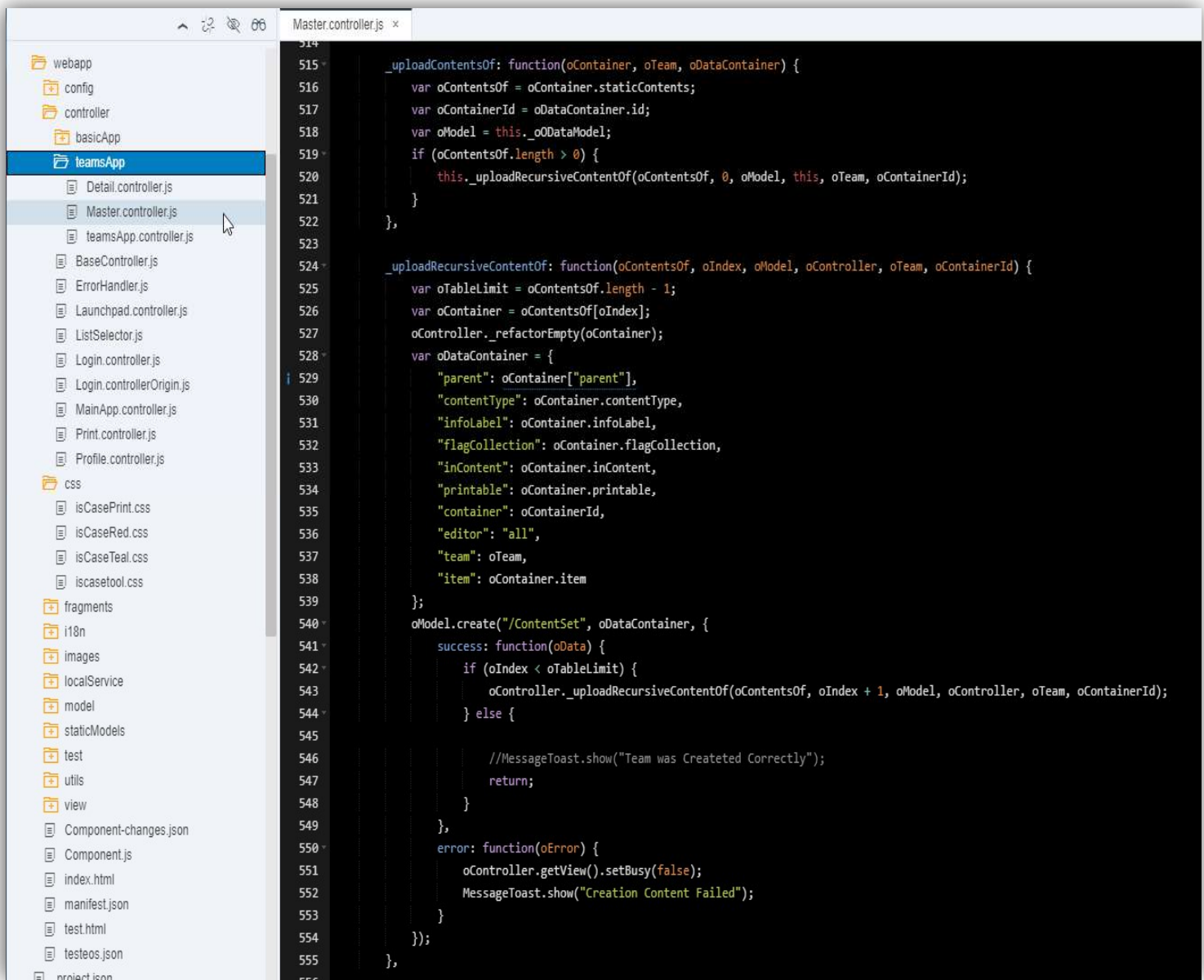
```

[Figura 13. Implementación del Alta de Entidades “Document” Front-end]

La Fase 2, se produce cuando un Miembro Administrador da de Alta un Equipo de Trabajo desde la UI, comprobando la existencia de los datos que en caso de no existir se dan de alta. Se Crean, los Contenidos Concretos de un Equipo de trabajo para cada tipo de Documento dado de alta en la BD (Entidades Content con su campo “inContent” vacío que rellenarán los miembros del Equipo de forma dinámica posteriormente).El alta se produce a partir de los ficheros “memModel.json”, “planModel.json” y “srsModel.json”



[Figura 14. Fragmento de Plantilla para Alta de Entidades “Item” y “Content” Front-end]



[Figura 15. Implementación del Alta de Entidades “Content” Front-end]

6.2.3 Interfaz de Usuario y su Funcionalidad Implementada

En este apartado, explica cómo se ha implementado la herramienta y presenta a la vez las funcionalidades ofrecidas a través de la Interfaz de Usuario.

IS Case Tool – Docs, implementa un punto de acceso común tanto a Miembros Administradores como a Miembros No administradores, a través de autenticación se realiza una diferenciación del tipo de usuario y se carga un Catálogo (**LaunchPad**) de funcionalidades adecuadas a cada tipo de usuario. Para ello cada usuario debe realizar un registro previo en la aplicación.

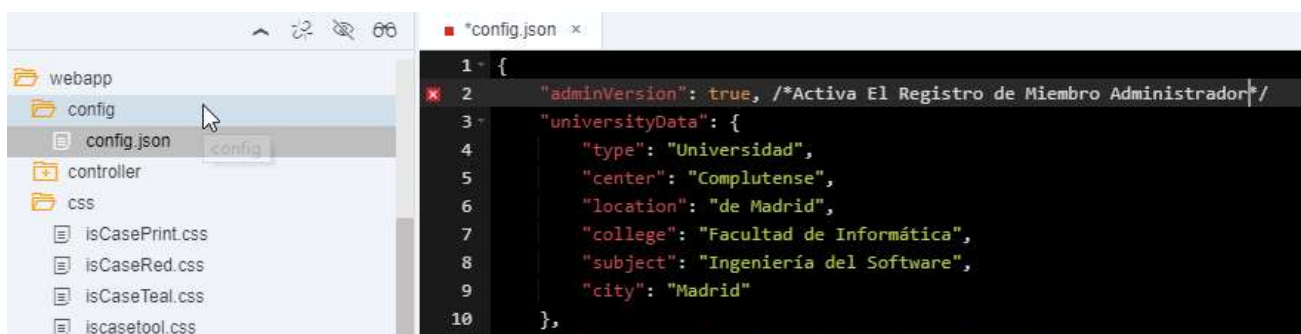
Para entrar en detalle, se clasifican las interfaces de usuario y sus funcionalidades de acuerdo al tipo de usuario:

6.2.3.1 UI y Funcionalidades para Miembro Administrador

- **Registro de Usuario)**

Es muy importante destacar que el Registro de Miembro/s Administrador/es, ha de realizarse previo a la puesta en producción de la herramienta, dado que la interfaz es común a todos los usuarios.

Para ello se ha dispuesto un fichero de configuración en el **Front-end** que permite visualizar el control de registro de un miembro administrador en la Aplicación ubicado en **“frontend/config/config.json”** y que además permite configurar algunos datos que se mostrarán en la Impresión Final de un Documento.

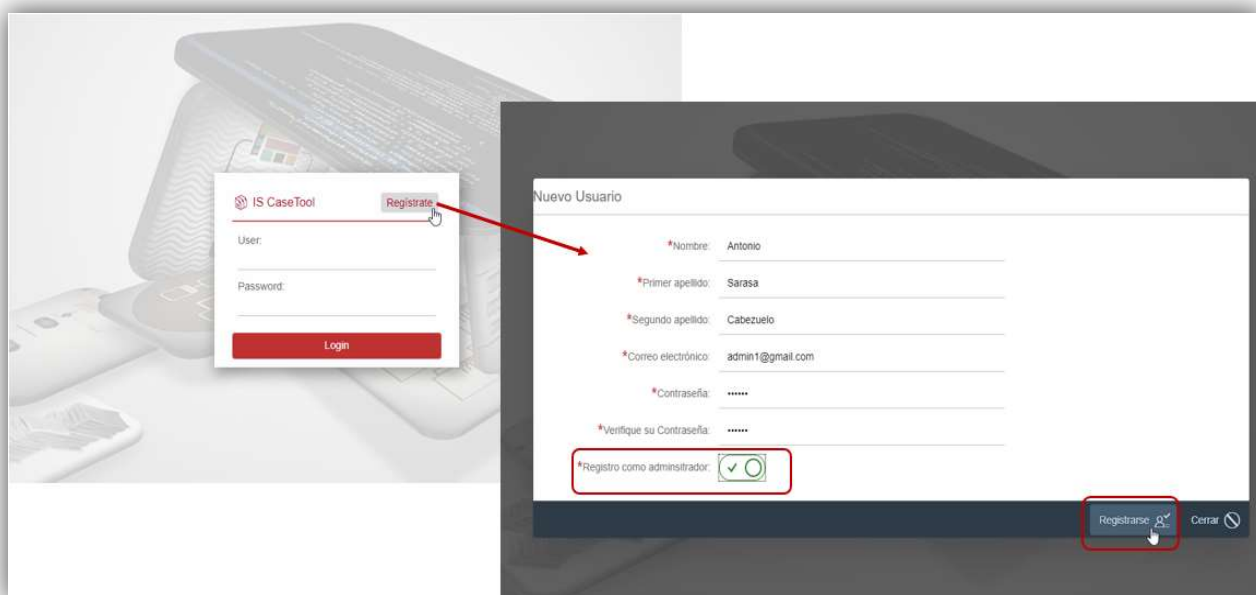


[Figura 16. Fichero JSON de configuración Admin]

A Partir de ahí y una vez arrancada la herramienta; el registro se realiza desde la vista principal **“Login”** utilizando el botón adecuado para este cometido **“Regístrate”**, se abrirá un formulario a rellenar.

Tras la validación de campos del formulario se activa la posibilidad de realizar el registro, se encripta la contraseña y se envía la petición de alta de usuario al Back-end.

Una vez creado el Usuario en la BD, se muestra un mensaje de confirmación.



[Figura 17. UI – Registro de Miembro Administrador]

```

Login.controller.js x
186     value1: oDataUser.oMailValue
187   });
188   var mParameters = {
189     filters: oFilters,
190     success: function(oData, oResponse) {
191       if (oData.results.length === 0) {
192         that._createNewUser(oDataUser);
193       } else {
194         MessageToast.show("El Usuario ha sido dado de alta con Anterioridad");
195       }
196     },
197     error: function(oError) {
198       MessageToast.show(oError.message + " -Miembro Set Service-read - " + oError.statusText);
199     }
200   };
201   oModel.read("/MiembroSet", mParameters);
202 },
203
204 /*Registro de Usuario una vez realizadas las comprobaciones*/
205 _createNewUser: function(oData) {
206   var that = this;
207   var oFinalData = {
208     "mail": oData.oMailValue,
209     "pass": oData.oPassValue,
210     "name": oData.oNameValue,
211     "lastName": oData.oLastnameValue,
212     "picture": oData.oPicValue,
213     "admin": oData.oAdminValue,
214     "director": oData.oDirectorValue,
215     "team": oData.oTeamValue
216   };
217
218   var oModel = this.getModel();
219   oModel.create("/MiembroSet", oFinalData, {
220     success: function(oDataRes) {
221       MessageToast.show(oDataRes.name + " " + oDataRes.lastName + "Has sido Dado de Alta, Ya Puedes Iniciar Sesión");
222       that.onCloseRegister();
223     },
224     error: function(oError) {
225       MessageToast.show("Technical error during creation");
226       that.onCloseRegister();
227     }
228   });
229 },

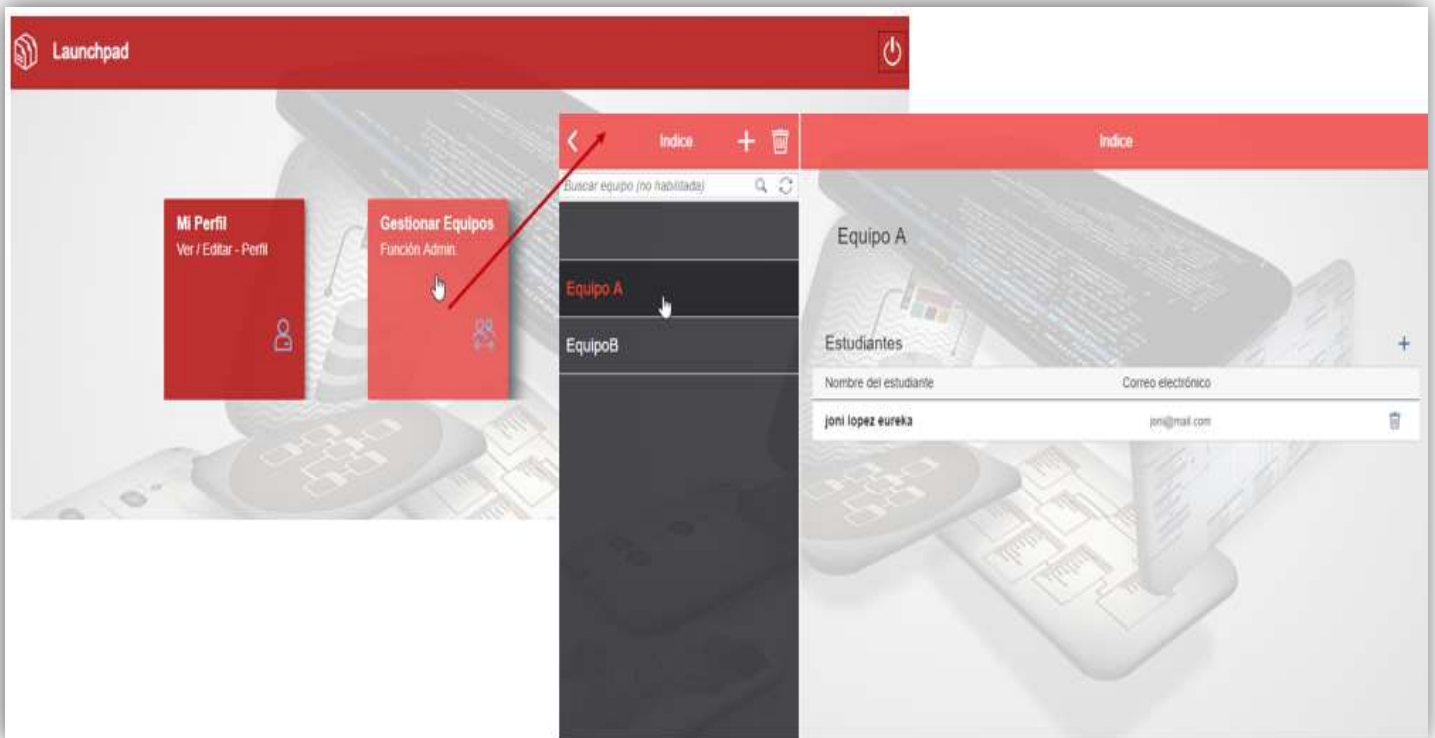
```

[Figura 18. Implementación – Registro de Miembro]

- **Acceso a Gestión de Equipos)**

Tras hacer un “LogIn” correcto en la pantalla principal, la herramienta comprueba el tipo de usuario (Admin) y **navega** al catálogo (Launchpad) donde se muestra el acceso a la Gestión de Equipos mediante un Tile.

Tras hacer Clic en el Tile “Gestionar Equipos”, la herramienta **Navega** a la vista de Gestión de Equipos, disponible sólo para Miembro/s Administrador/es.



[Figura 19. UI – Acceso a Gestión de Equipos]

```
Launchpad.controller.js x
192
193   onPressApp: function(oEvent) {
194       var oTheme = this._theme;
195       var oGrid = this.getView().byId("superiorContainer");
196       var oTiles = oGrid.getContent();
197       var oSelectedTile = oEvent.getSource();
198       var oClass = oTheme.getPrincipalClass();
199       var oActualPosStyle = 0;
200
201       for (var i = 0; i < oTiles.length; i++) {
202           var toSearch = oClass + i.toString();
203           if (oSelectedTile.hasStyleClass(toSearch)) {
204               oActualPosStyle = i;
205               break;
206           }
207       }
208       oTheme.setSelectedPos(oActualPosStyle);
209
210       var oObj = oEvent.getSource().getBindingContext().getObject();
211       var oRouter = this.getRouter();
212
213       if (oObj.credential === "US") { // si es app de Usuario saltamos a basicApp
214           if (oObj.docType === "SRS") { // de momento solo se activa el SRS
215               oRouter.navTo("basicApp", {
216                   id: encodeURIComponent(0),
217                   docType: encodeURIComponent(oObj.docType)
218               }, true);
219           }
220       }
221
222       if (oObj.credential === "ALL") { // si es app de credencial ALL todos tienen acceso
223           if (oObj.docType === "PROFILE") {
224               oRouter.navTo("profile", {
225                   docType: encodeURIComponent(oObj.docType)
226               }, true);
227           }
228       }
229
230       if (oObj.credential === "AD") { //si es App de Admin saltamos a la app de Administración
231           oRouter.navTo("teamsApp", {
232               docType: encodeURIComponent(oObj.docType)
233           }, true);
234       }
```

[Figura 20. Implementación – Acceso a Gestión de Equipos

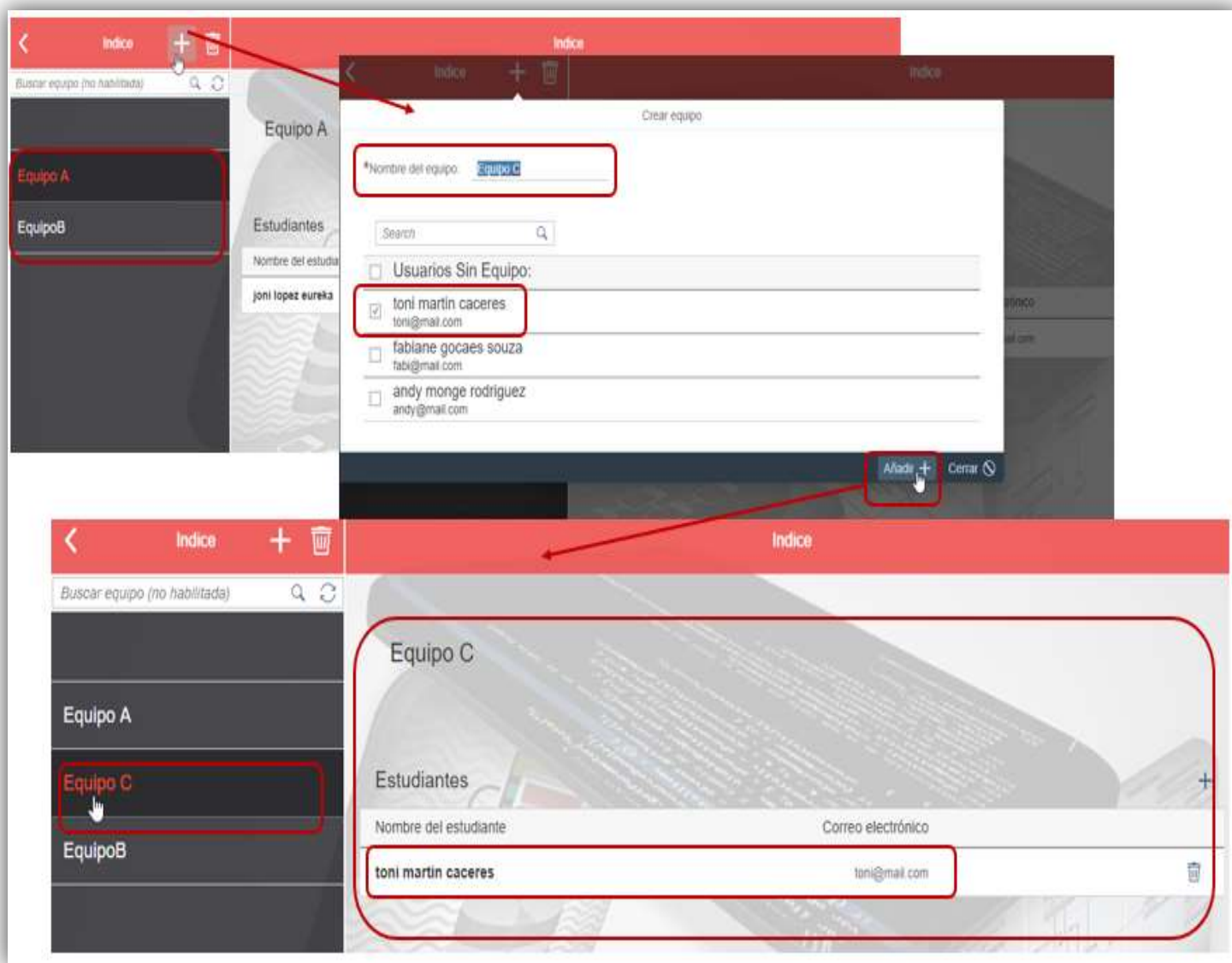
- **Alta de un Equipo)**

Desde la Vista de Gestión de Equipos, haciendo clic sobre el botón “Agregar Equipo” ubicado en el menú de la izquierda que muestra los equipos actualmente creados; la herramienta presenta un **PopOver** en el que se debe definir el Nombre del Nuevo Equipo y los Miembros Integrantes (al menos uno) que lo componen.

Tras validar lo anterior, se activa la posibilidad de Crear un Nuevo Equipo mediante el botón “Añadir”.

El Usuario, hace clic en el Botón Añadir, ordenando el envío de los datos para la creación del nuevo Equipo al Back-end.

La herramienta, atacando al back-end; da de alta el Equipo, Actualiza el campo “team” de todos sus miembros y crea los contenidos estáticos del equipo relacionados con el SRS (único documento activo). Al terminar se muestra un mensaje de confirmación y se actualiza la Lista de Equipos mostrada en la vista.



[Figura 21. UI – Alta de un Equipo]

```

387 },
388
389 onCreateTeam: function(oEvent) {
390     this._correctCreatedUsers = [];
391     var that = this;
392     var oView = this.getView();
393     oView.setBusy(true);
394     var oInputContainer = this._oPopoverCreate.getContainer();
395     var oInputValue = oInputContainer.getItems()[1].getValue();
396     var oModel = this._oDataModel;
397     var oTable = this._oPopoverCreate.getContent()[0].getTable();
398     var oSelectedCollection = oTable.getSelectedItems();
399
400     var oDataTeam = {
401         "name": oInputValue,
402         "director": "0"
403     };
404
405     oModel.create("/TeamSet", oDataTeam, {
406         success: function(oData) {
407             that._recursiveUpdateUsers(oSelectedCollection, 0, oModel, oController, oTeam);
408         },
409         error: function(oError) {
410             MessageToast.show("Update Failed");
411         }
412     });
413
414     //this._recursiveUpdateUsers(oSelectedCollection, 0, oModel, oController, oTeam);
415     this._onCloseCreateTeam(null);
416 },
417
418 _recursiveUpdateUsers: function(oSelectedCollection, oIndex, oModel, oController, oTeam) {
419     var oTableLimit = oSelectedCollection.length - 1;
420     var oItem = oSelectedCollection[oIndex];
421     var oContext = oItem.getBindingContext().getObject();
422     var oPath = oItem.getBindingContext().getPath();
423     var oData = {
424         "team": oTeam
425     };
426     oModel.update(oPath, oData, {
427         success: function(oDataRes) {
428             if (oIndex < oTableLimit) {
429                 oController._recursiveUpdateUsers(oSelectedCollection, oIndex + 1, oModel, oController, oTeam);
430             } else {
431                 oController._allMembersUpdated(oTeam);
432                 return;
433             }
434         },
435         error: function(oError) {
436             oController.getView().setBusy(false);
437             MessageToast.show("Error Members Up Failed");
438         }
439     });
440 },
441
442 _allMembersUpdated: function(oTeam) { // params: team; editor hay que poner a "all", "nobody"
443     var oFinalDocCollection = null;
444     var oModelApps = this.getModel("Apps");
445     var oCollectionToUpload = oModelApps.getData().Documents;
446     if (oCollectionToUpload.length > 0) {
447         oFinalDocCollection = oCollectionToUpload;
448     }
449     if (oFinalDocCollection) {
450         for (var i = 0; i < oFinalDocCollection.length; i++) {
451             var oCurrentApp = oFinalDocCollection[i];
452             if (oCurrentApp.credential === "US" && oCurrentApp.modelName && oCurrentApp.docType) {
453                 this._bufferingContentsModel(oTeam, oCurrentApp.modelName);
454             }
455         }
456     } else {
457         //Incluir mensaje de Error en consulta de Apps
458         MessageToast.show("Error buffering static model failed");
459     }
460 }

```

[Figura 22. Implementación– Alta de un Equipo]

- **Modificar Miembros de un Equipo)**

Desde la Vista de Gestión de Equipos, es posible agregar o Eliminar Miembros de un Equipo mediante los Botones dispuestos para ello en la parte derecha de la Vista donde se muestran los detalles de un Equipo.

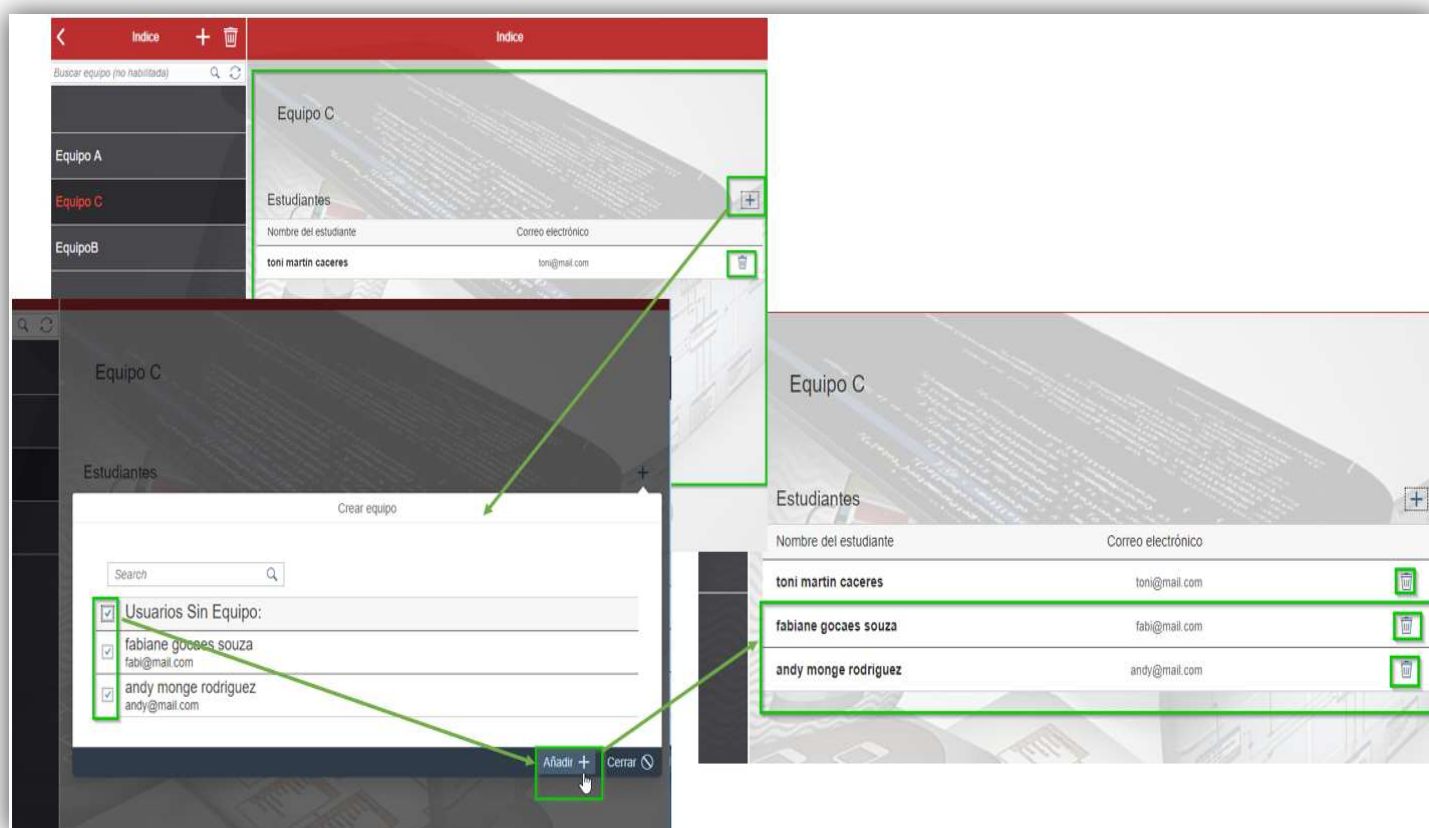
Para Agregar un Miembro al Equipo, el usuario debe utilizar el botón de “Añadir Miembro”, que abre un PopOver similar al de alta de Equipos, donde se muestran los usuarios seleccionables que aún no pertenecen a ningún Equipo. Tras seleccionar uno o más usuarios se activa la posibilidad de Agregar más miembros al Equipo mediante el botón “Añadir”.

El Usuario, hace clic en el Botón Añadir, ordenando el envío de los datos para agregación del nuevo Miembro al Equipo que se pide al Back-end.

La herramienta, atacando al back-end; Actualiza el campo “team” de todos los miembros Nuevos seleccionados. Al terminar se muestra un mensaje de confirmación y se actualiza la Lista de Miembros del Equipo mostrada en la vista.

Para Eliminar un Miembro del Equipo, simplemente se debe hacer uso del botón “Eliminar” mostrado junto a la información de cada Miembro, ordenando el envío de los datos para la exclusión del Miembro seleccionado de su Equipo, que se pide al Back-end.

La herramienta, atacando al back-end; Actualiza el campo “team” del miembro Excluido dejándolo sin Valor. Al terminar se muestra un mensaje de confirmación y se actualiza la Lista de Miembros del Equipo mostrada en la vista.



[Figura 23. UI – Modificar Miembros de un Equipo]

```

246 oncreateTeam: function(oEvent) { // calls when the Admin confirm to add a Member
247     this._CorrectCreatedUsers = [];
248     var that = this;
249     var oModel = this._oDataModel;
250     var oTable = this._oPopoverCreate.getContent()[0].getContent()[0];
251     var oSelectedCollection = oTable.getSelectedItems();
252     var oTeamId = this.getView().getBindingContext().getObject().id;
253     this._recursiveUpdateUsers(oSelectedCollection, 0, oModel, this, oTeamId);
254 },
255
256 _recursiveUpdateUsers: function(oSelectedCollection, oIndex, oModel, oController, oTeam) {
257     var oTableLimit = oSelectedCollection.length - 1;
258     var oItem = oSelectedCollection[oIndex];
259     var oPath = oItem.getBindingContext().getPath();
260     var oData = {
261         "team": oTeam
262     };
263     oModel.update(oPath, oData, {
264         success: function(oDataRes) {
265             if (oIndex < oTableLimit) {
266                 oController._recursiveUpdateUsers(oSelectedCollection, oIndex + 1, oModel, oController, oTeam);
267             } else {
268                 oController._allMembersUpdated(oTeam);
269                 return;
270             }
271         },
272         error: function(oError) {
273
274         }
275     });
276 },
277
278 _allMembersUpdated: function(oTeam) { // params: team,; editor hay que poner a "all", "nobody"
279     var oTable = this.getView().byId("lineItemsList");
280     if (this._oPopoverCreate) {
281         this.onCloseCreateTeam();
282     }
283     oTable.getBinding("items").refresh();
284 },

```

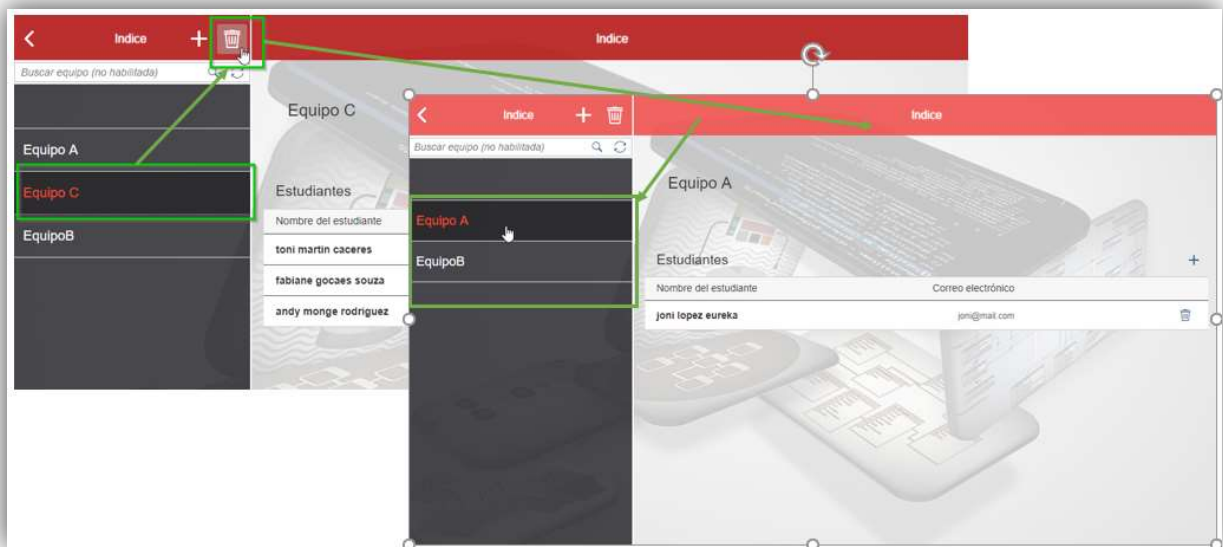
[Figura 24. Implementación– Modificar Miembros de un Equipo]

- **Baja de un Equipo)**

Desde la Vista de Gestión de Equipos, haciendo clic sobre el botón “Eliminar Equipo” ubicado en el menú de la izquierda que muestra los equipos actualmente creados.

El Usuario, hace clic en el Botón mencionado, ordenando el envío de los datos para el borrado del Equipo al Back-end.

La herramienta, atacando al back-end; borra todos los contenidos del equipo relacionados con el SRS (único documento activo), Actualiza el campo “team” de todos sus miembros y elimina el Equipo de la BD. Al terminar se muestra un mensaje de confirmación y se actualiza la Lista de Equipos mostrada en la vista.



[Figura 25.UI – Baja de un Equipo]

```
_deleteOneEntity: function(sPath, fnSuccess, fnFailed) {
    var oPromise = new Promise(function(fnResolve, fnReject) {
        this._oDataModel.setUseBatch(false);
        this._oDataModel.remove(sPath, {
            success: fnResolve,
            error: fnReject,
            async: true
        });
    }).bind(this);
    oPromise.then(fnSuccess, fnFailed);
    return oPromise;
}

});
```

[Figura 26. Implementación – Baja de un Equipo]

6.2.3.2 UI y Funcionalidades para Miembro (No Administrador)

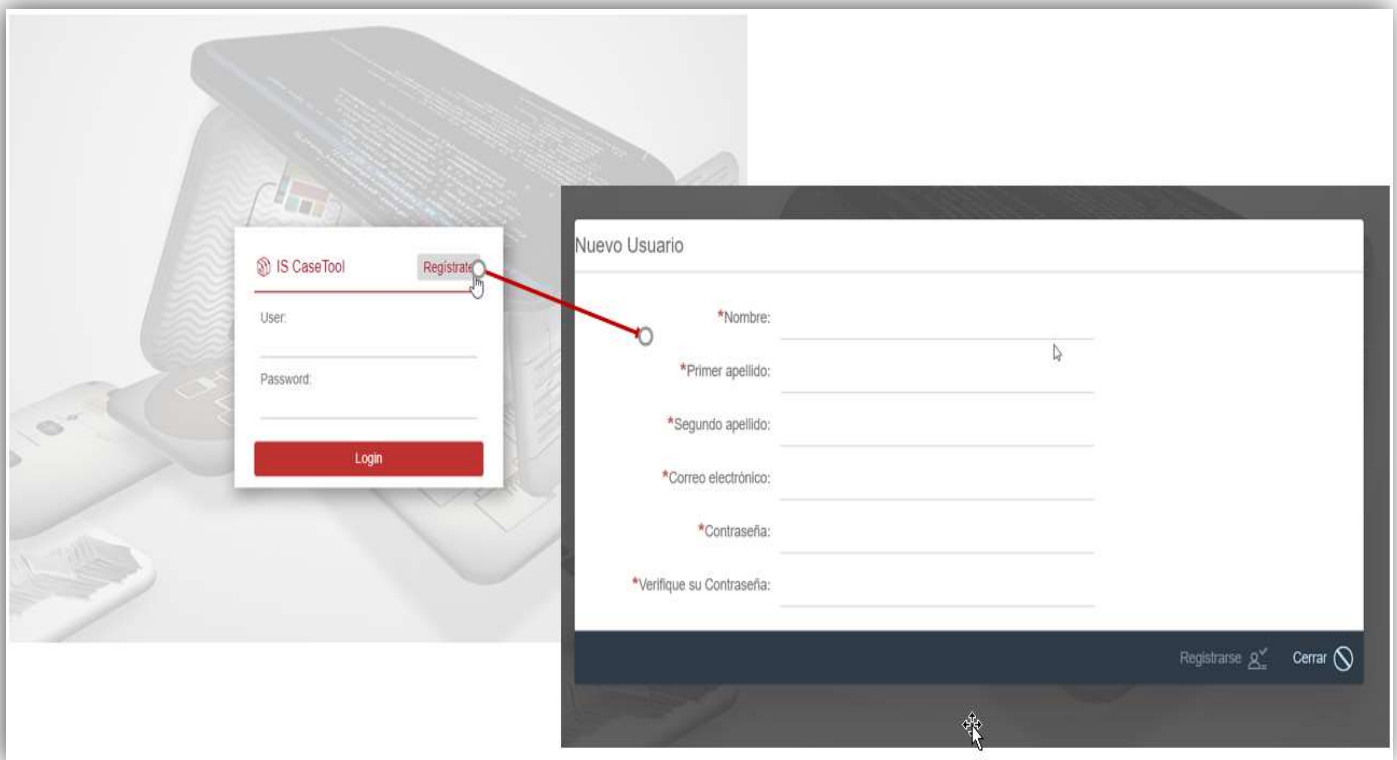
- **Registro de Usuario)**

Tras configurar el fichero “**config. json**” y poner en producción la herramienta, un Miembro será capaz de realizar el registro de usuario.

El registro se realiza desde la vista principal “**Login**” utilizando el botón adecuado para este cometido “**Regístrate**”, se abrirá un formulario a rellenar.

Tras la validación de campos del formulario se activa la posibilidad de realizar el registro, se encripta la contraseña y se envía la petición de alta de usuario al Back-end.

Una vez creado el Usuario en la BD, se muestra un mensaje de confirmación.

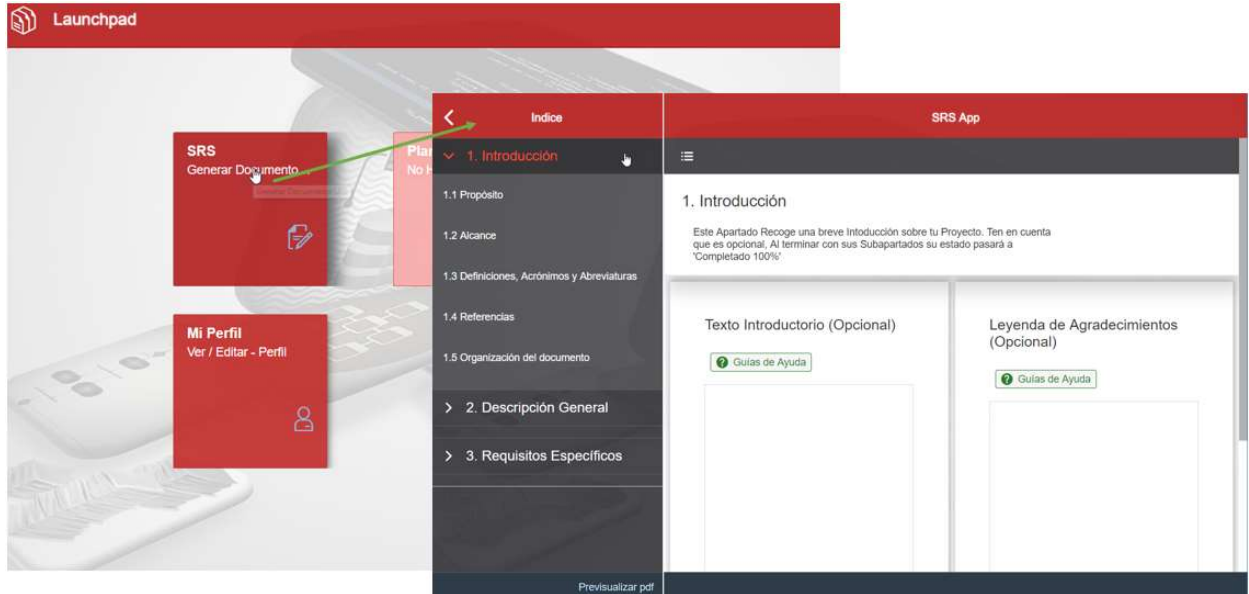


[Figura 27. UI – Registro de Miembro No Administrador]

- **Acceso a un Documento concreto (SRS en este caso)**

Tras hacer un “Login” correcto en la pantalla principal, la herramienta comprueba el tipo de usuario (No Admin) y **navega** al catálogo (Launchpad) donde se muestra el acceso al Documento SRS mediante un Tile.

Tras hacer Clic en el Tile “SRS”, la herramienta **Navega** a la vista de edición del SRS, disponible sólo para Miembro/s No Administrador/es.



[Figura 28. UI – Acceso a SRS desde LaunchPad]

```

Launchpad.controller.js x
192
193 -   onPressApp: function(oEvent) {
194     var oTheme = this._theme;
195     var oGrid = this.getView().byId("superiorContainer");
196     var oTiles = oGrid.getContent();
197     var oSelectedTile = oEvent.getSource();
198     var oClass = oTheme.getPrincipalClass();
199     var oActualPosStyle = 0;
200
201 -   for (var i = 0; i < oTiles.length; i++) {
202     var toSearch = oClass + i.toString();
203     if (oSelectedTile.hasStyleClass(toSearch)) {
204       oActualPosStyle = i;
205       break;
206     }
207   }
208   oTheme.setSelectedPos(oActualPosStyle);
209
210   var oObj = oEvent.getSource().getBindingContext().getObject();
211   var oRouter = this.getRouter();
212
213 -   if (oObj.credential === "US") { // si es app de Usuario saltamos a basicApp
214     if (oObj.docType === "SRS") { // de momento solo se activa el SRS
215       oRouter.navTo("basicApp", {
216         id: encodeURIComponent(0),
217         docType: encodeURIComponent(oObj.docType)
218       }, true);
219     }
220   }

```

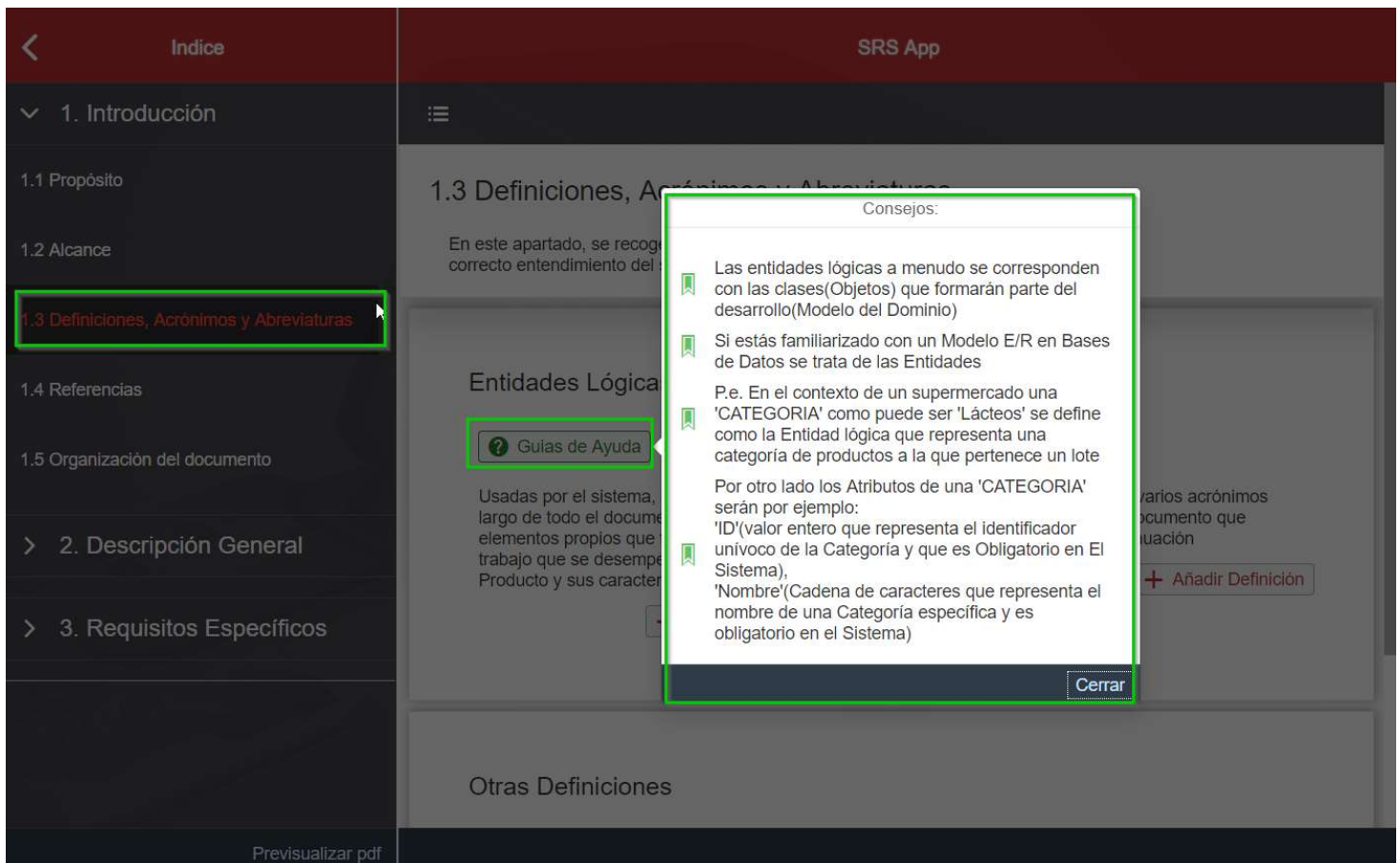
[Figura 29. Implementación – Acceso a SRS desde LaunchPad]

- **Visualización de Guías de Ayuda)**

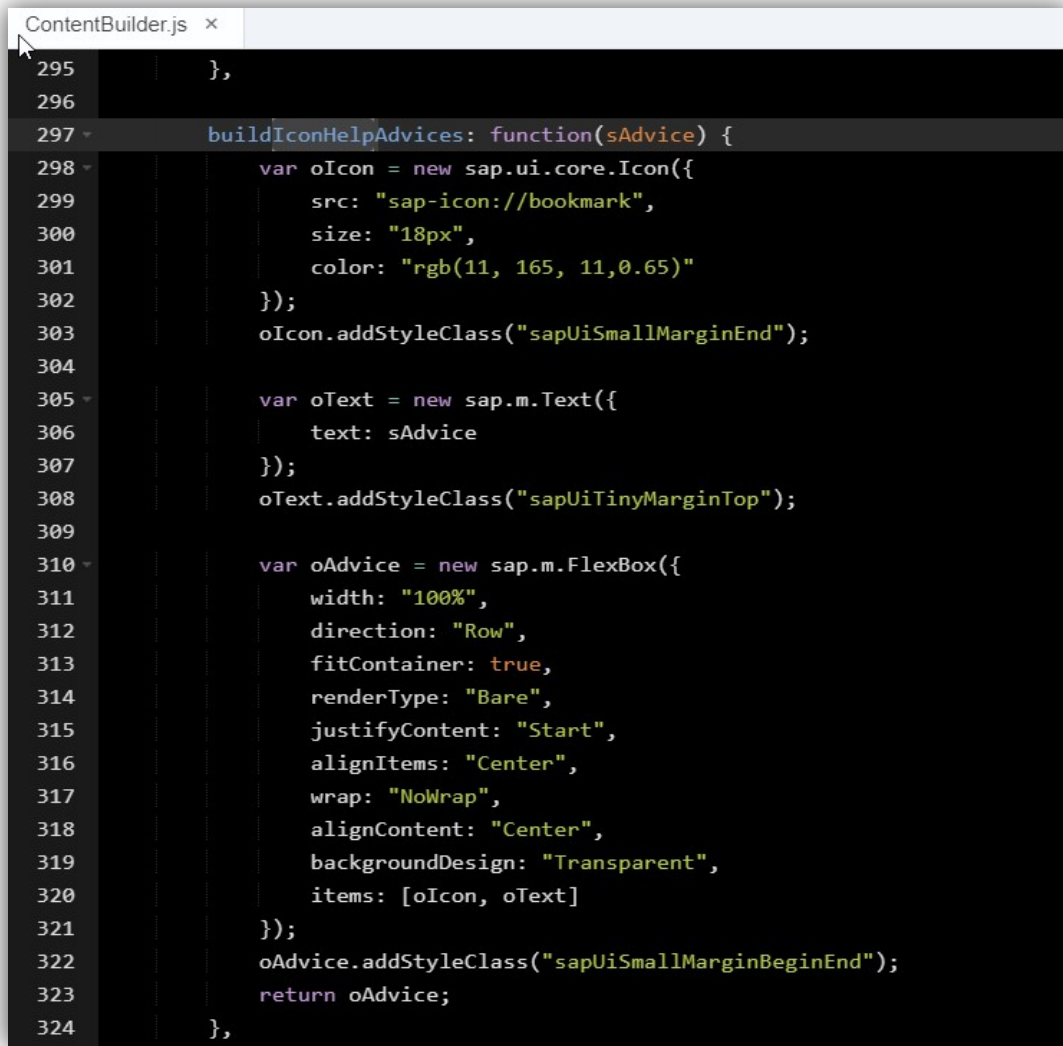
Una vez se accede a la vista del SRS, un usuario es capaz de visualizar las guías de ayuda en todos y cada uno de los apartados que elija en el menú de la izquierda. Dichas ayudas se crean mediante un Contenido almacenado por plantilla.

Consisten en un botón “Guías de ayuda” que abre un PopOver con consejos y ejemplos para rellenar el apartado correspondiente.

La herramienta busca el constructor de ayuda, accede a la información oportuna y muestra los consejos.



[Figura 30. UI –Ver Guías de Ayuda]



```
ContentBuilder.js x
295     },
296
297     buildIconHelpAdvices: function(sAdvice) {
298         var oIcon = new sap.ui.core.Icon({
299             src: "sap-icon://bookmark",
300             size: "18px",
301             color: "rgb(11, 165, 11,0.65)"
302         });
303         oIcon.addClass("sapUiSmallMarginEnd");
304
305         var oText = new sap.m.Text({
306             text: sAdvice
307         });
308         oText.addClass("sapUiTinyMarginTop");
309
310         var oAdvice = new sap.m.FlexBox({
311             width: "100%",
312             direction: "Row",
313             fitContainer: true,
314             renderType: "Bare",
315             justifyContent: "Start",
316             alignItems: "Center",
317             wrap: "NoWrap",
318             alignContent: "Center",
319             backgroundDesign: "Transparent",
320             items: [oIcon, oText]
321         });
322         oAdvice.addClass("sapUiSmallMarginBeginEnd");
323         return oAdvice;
324     },
```

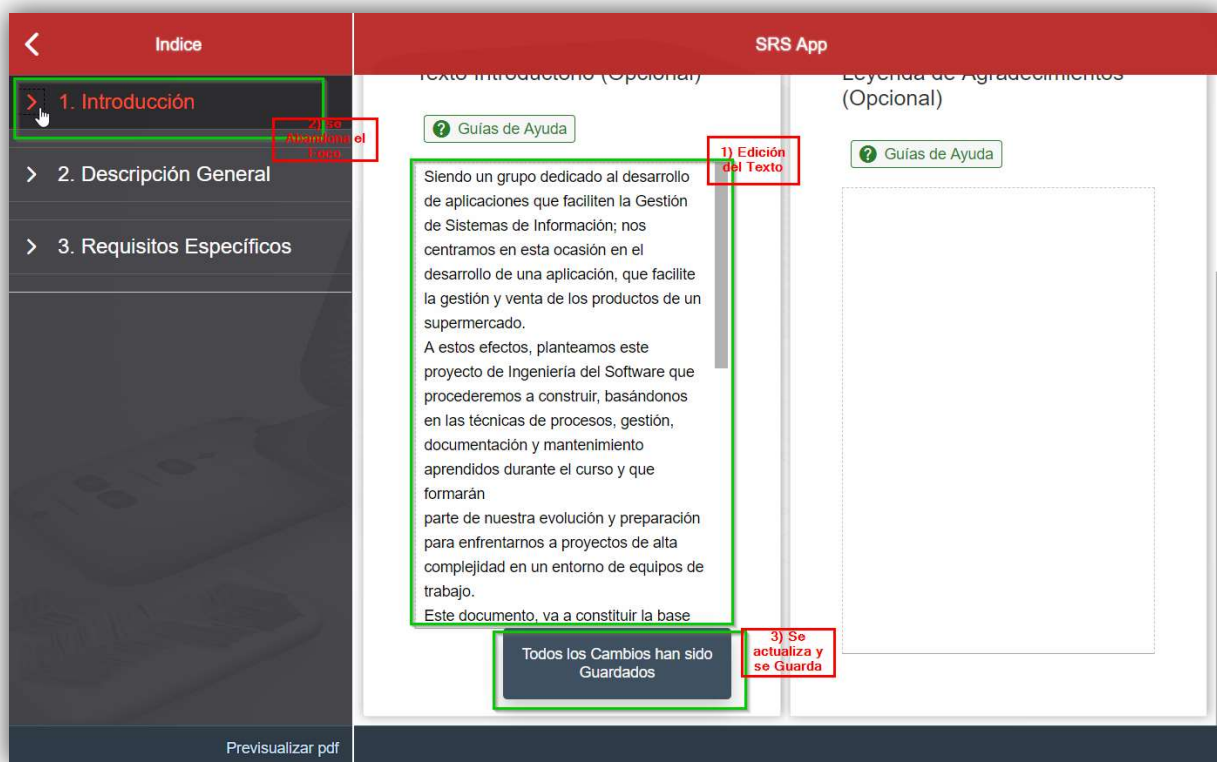
[Figura 31. Implementación –Ver Guías de Ayuda]

- **Edición de Apartados del Documento y Auto Guardado)**

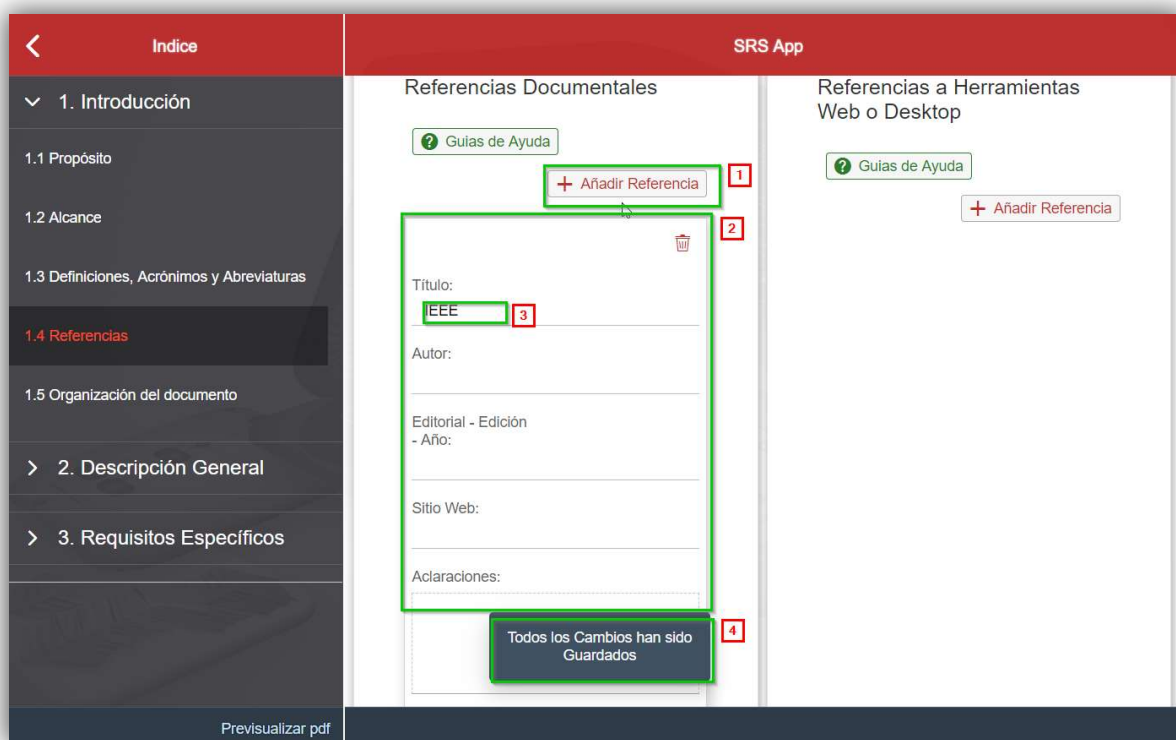
Una vez se accede a la vista del SRS, un usuario es capaz de editar el contenido en todos y cada uno de los apartados que elija en el menú de la izquierda. Dicho contenido se rellena de forma distinta personalizada e intuitiva en cada apartado, utilizando TextAreas, agregación de módulos mediante botones, agregación de imágenes, etc.

Al abandonar el Foco del elemento editado, la herramienta busca el constructor de contenido adecuado, añade la información rellena por el usuario sobre el Modelo y envía la información al

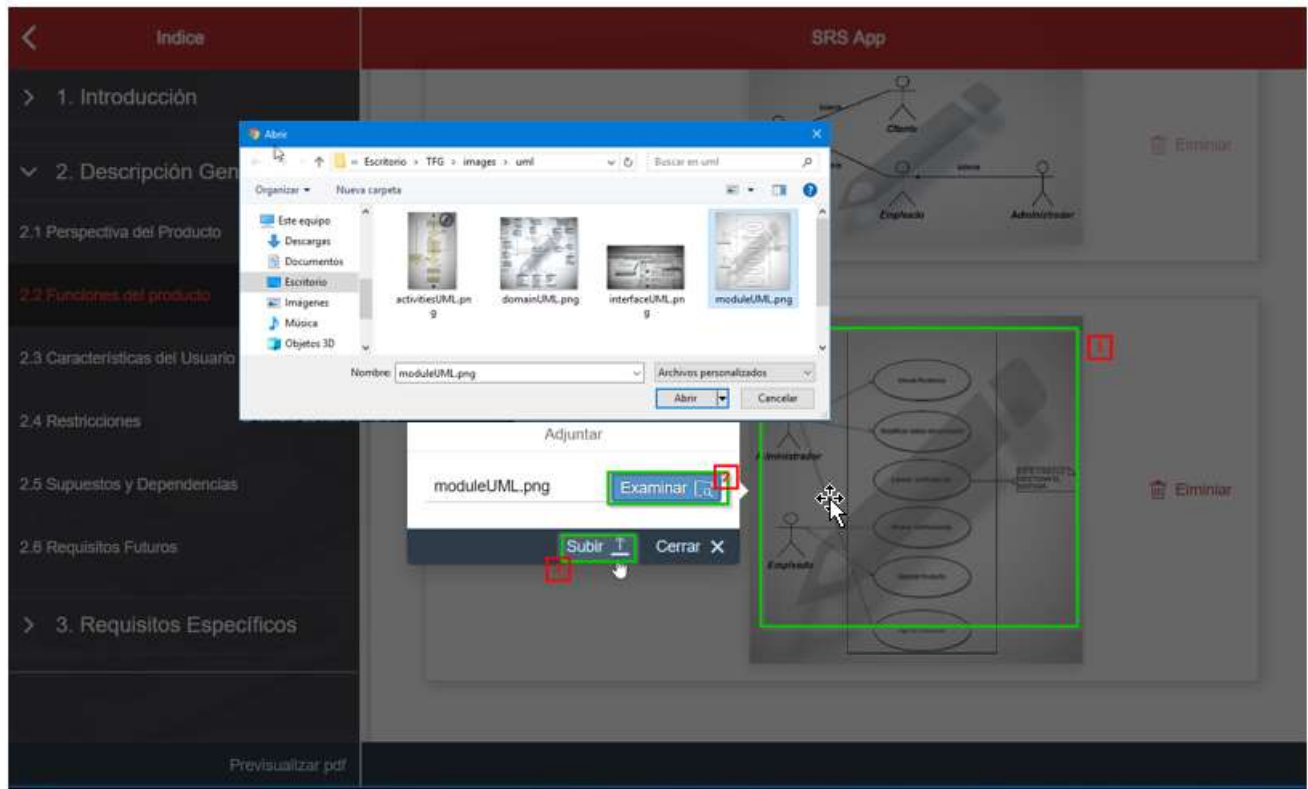
back-end. Se almacena la información y se muestra un Mensaje afirmando el Guardado automático de la misma y se reconstruyen los elementos de la UI de ser necesario



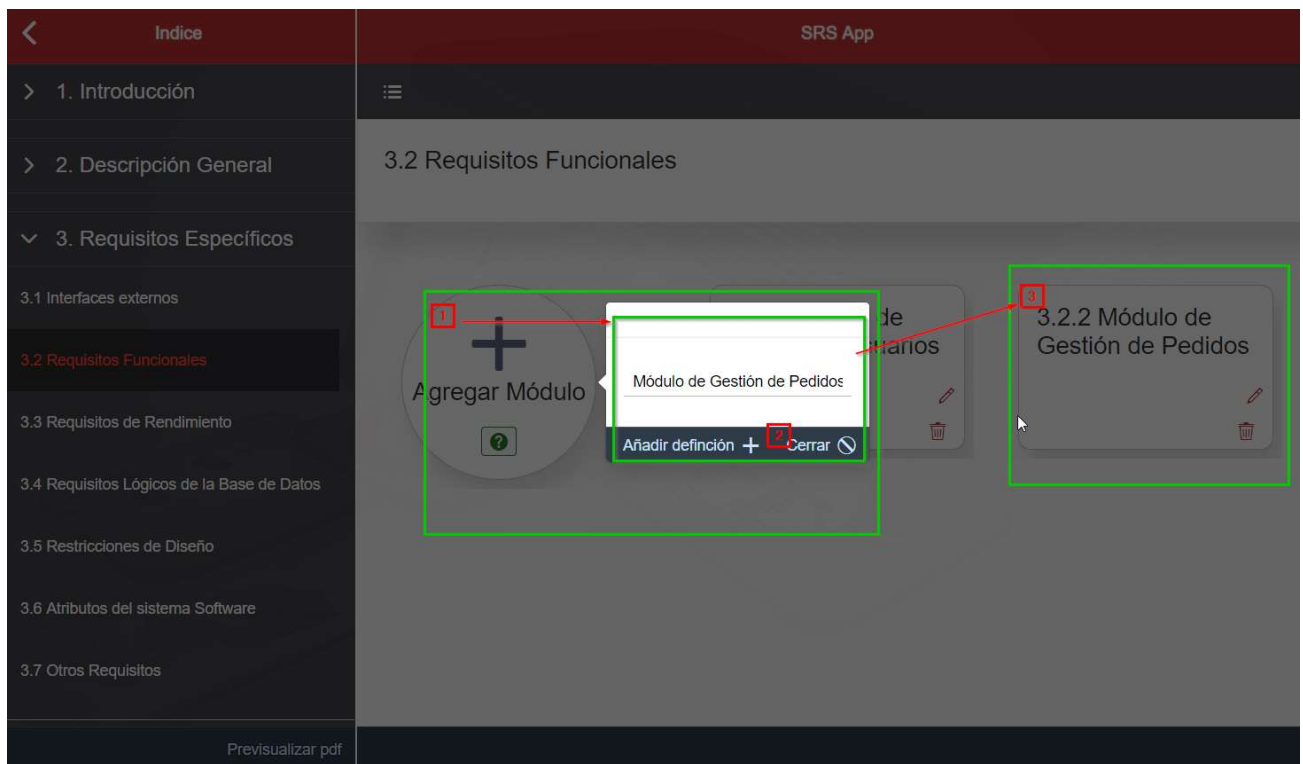
[Figura 32. UI – Edición de un TextArea]



[Figura 33. UI – Edición de un Módulo Compuesto]



[Figura 34. UI – Edición de una Imagen]



[Figura 35. UI – Agregación de Subpartados de 2do Nivel]

```

    _onChangeInputContent: function(oSource, oAdditionalSource) {
        this._changeInput = false;
        this.getOwnerComponent().getRootControl().setBusy(true);
        var that = this;
        var oBinding = oSource.getBindingContext();
        if (!oBinding && oAdditionalSource) {
            oBinding = oAdditionalSource.getBindingContext();
        }
        var oContextObject = oBinding.getObject();
        var oContextPath = oBinding.getPath();
        var oNewValue = oSource.getValue();
        if (oNewValue.length === 0) {
            oNewValue = "\n";
        }
        var oData = {
            "infoLabel": oContextObject.infoLabel,
            "inContent": oNewValue
        };

        this._oODataModel.update(oContextPath, oData, {
            success: function(oDataIn) {
                //that._loadContent();
                that._pageBase.scrollToElement(oSource);
                that.getOwnerComponent().getRootControl().setBusy(false);
                MessageToast.show("Todos los Cambios han sido Guardados");
            },
            error: function(oError) {
                that.getOwnerComponent().getRootControl().setBusy(false);
                MessageToast.show("Fallo al guardar el último Cambio");
            }
        });
    }

```

[Figura 36. Implementación – Edición de Contenidos]

Detail_controller.js x

```

427     var oKeyPath = oEntityName + "(" + oData.results[i].id + ")";
428     var oSingleContent;
429     switch (oData.results[i].contentType) {
430         case "textArea":
431             oSingleContent = that._ContentBuilder.buildTextAreaType(oData.results[i], that._myBlockSize, true);
432             oSingleContent.bindElement(oKeyPath);
433             break;
434         case "shortTextArea":
435             oSingleContent = that._ContentBuilder.buildTextAreaType(oData.results[i], that._myBlockSize, false);
436             oSingleContent.bindElement(oKeyPath);
437             break;
438         case "iconHelp":
439             oSingleContent = that._ContentBuilder.buildIconHelpType(oData.results[i]);
440             oSingleContent.bindElement(oKeyPath);
441             break;
442         case "input":
443             oSingleContent = that._ContentBuilder.buildInputType(oData.results[i]);
444             oSingleContent.bindElement(oKeyPath);
445             break;
446
447         case "infoBox":
448             oSingleContent = that._ContentBuilder.buildInfoBoxType(oData.results[i]);
449             break;
450
451         case "butonAddBox2":
452             oSingleContent = that._ContentBuilder.buildButtonAddDefinitionType(oData.results[i], true,
453                 oContainerBlock);
454             break;
455

```

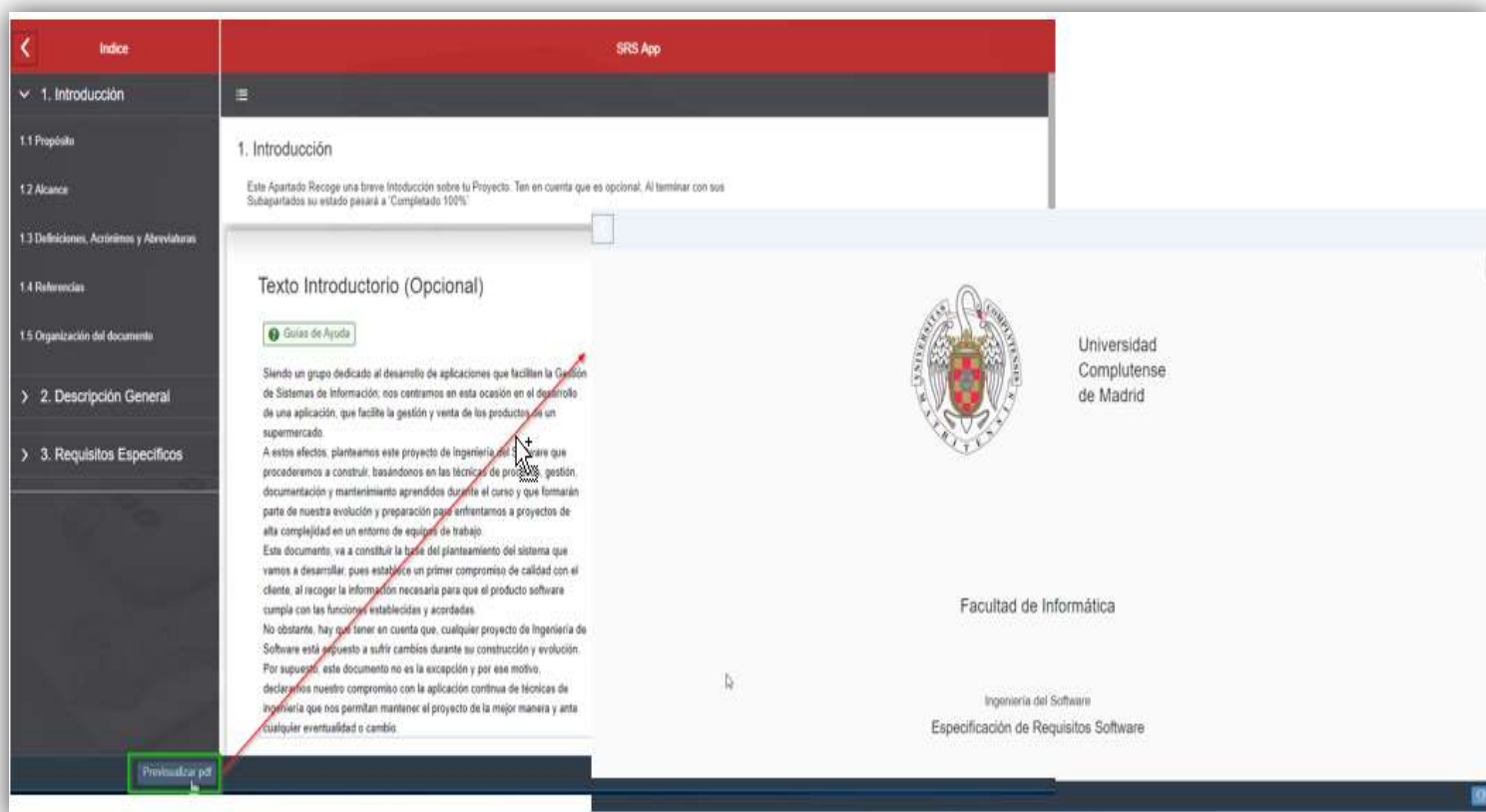
[Figura 37. Implementación – Construcción de Contenidos por tipo]

- **Previsualización del Documento Final)**

Una vez se accede a la vista del SRS, un usuario es capaz de navegar a una vista con la previsualización del documento final que será impreso.

La acción se realiza al hacer clic en el botón “Previsualizar PDF”.

La herramienta, recoge y organiza: todos los datos de equipo, índice y contenidos del Documento del Equipo al que pertenece el Usuario con sesión abierta. Y finalmente, construye la Vista con el formato de impresión.



[Figura 38. UI– Previsualización de PDF]


```

Print.controller.js x
230 _loadContents: function() {
231     var that = this;
232     var oView = this.getView();
233     var oContentPrincipalBox = this.getView().byId("contentPrincipal");
234     var oFilter = [new Filter("doc", FilterOperator.EQ, utilFuncs.addQuotMarks(this._oDocType))];
235
236     oContentPrincipalBox.bindAggregation("items", {
237         path: "/ItemSet",
238         filters: oFilter,
239         factory: that._itemsFactory.bind(that),
240         events: {
241             change: function() {
242             }
243         }
244     });
245 });
246
247 },
248
Print.controller.js x
152
153 _loadIndex: function() {
154     var that = this;
155     var oIndexPrincipalBox = this.getView().byId("indexPrincipal");
156     var oFilter = [new Filter("doc", FilterOperator.EQ, utilFuncs.addQuotMarks(this._oDocType))];
157
158     oIndexPrincipalBox.bindAggregation("items", {
159         path: "/ItemSet",
160         filters: oFilter,
161         factory: that._indexFactory.bind(that),
162         events: {
163             change: that._loadContents.bind(that)
164         }
165     });
166 },
167
Print.controller.js x
70
71 _initStaticTexts: function(sDocType, sTeam) {
72     var oBundle = this.getResourceBundle();
73     var oView = this.getView();
74     var oDocTitle = oView.byId("docTitle");
75     var oTeamTitle = oView.byId("teamTitle");
76     var oLocationTitle = oView.byId("locationYear");
77     var sDoc = "";
78     var oModel = this._oDataModel;
79     var ssTeam = "";
80
81     var mParameters = {
82         success: function(oData, oResponse) {
83             if (oData.results.length > 0) {
84                 for (var i = 0; i < oData.results.length; i++) {
85                     var oActualTeam = oData.results[i].id;
86                     if (oActualTeam === sTeam) {
87                         ssTeam = oData.results[i].name;
88                     }
89                 }
90
91                 var sLocation = oLocationTitle.getText() + " - " + (new Date().getFullYear()).toString();
92                 if (sDocType === "SRS") {
93                     sDoc = oBundle.getText("SRSTitle");
94                 }
95                 if (sDocType === "PLAN") {
96                     sDoc = oBundle.getText("PLANTitle");
97                 }
98                 if (sDocType === "MEM") {
99                     sDoc = oBundle.getText("MEMTitle");
100                 }
101             }
102         }
103     };
104
105     oModel.read(sLocation, mParameters);
106
107     oDocTitle.setText(sDoc);
108     oTeamTitle.setText(ssTeam);
109
110     this._onPrintMatched(oEvent);
111 }
112
113 _onPrintMatched: function(oEvent) {
114     var oView = this.getView();
115     oView.setBusy(true);
116     var oParameter = oEvent.getParameter("arguments");
117     for (var value in oParameter) {
118         oParameter[value] = decodeURIComponent(oParameter[value]);
119     }
120     this._oDocType = oParameter.docType;
121
122     var oMembersTable = this.getView().byId("membersTable");
123
124     var oSession = this.getOwnerComponent().getSession();
125     var oTeam = utilFuncs.addQuotMarks(oSession.team);
126
127     this._initStaticTexts(this._oDocType, oSession.team);
128
129     var oFilter = [new Filter("team", FilterOperator.EQ, oTeam)];
130
131     var oTemplate = new sap.m.ColumnListItem({
132         cells: [
133             new sap.m.Text({
134                 text: "{name} {lastName}"
135             })
136         ]
137     });
138
139     oMembersTable.bindItems({
140         path: "/MiembroSet",
141         filters: oFilter,
142         template: oTemplate
143     });
144 }

```

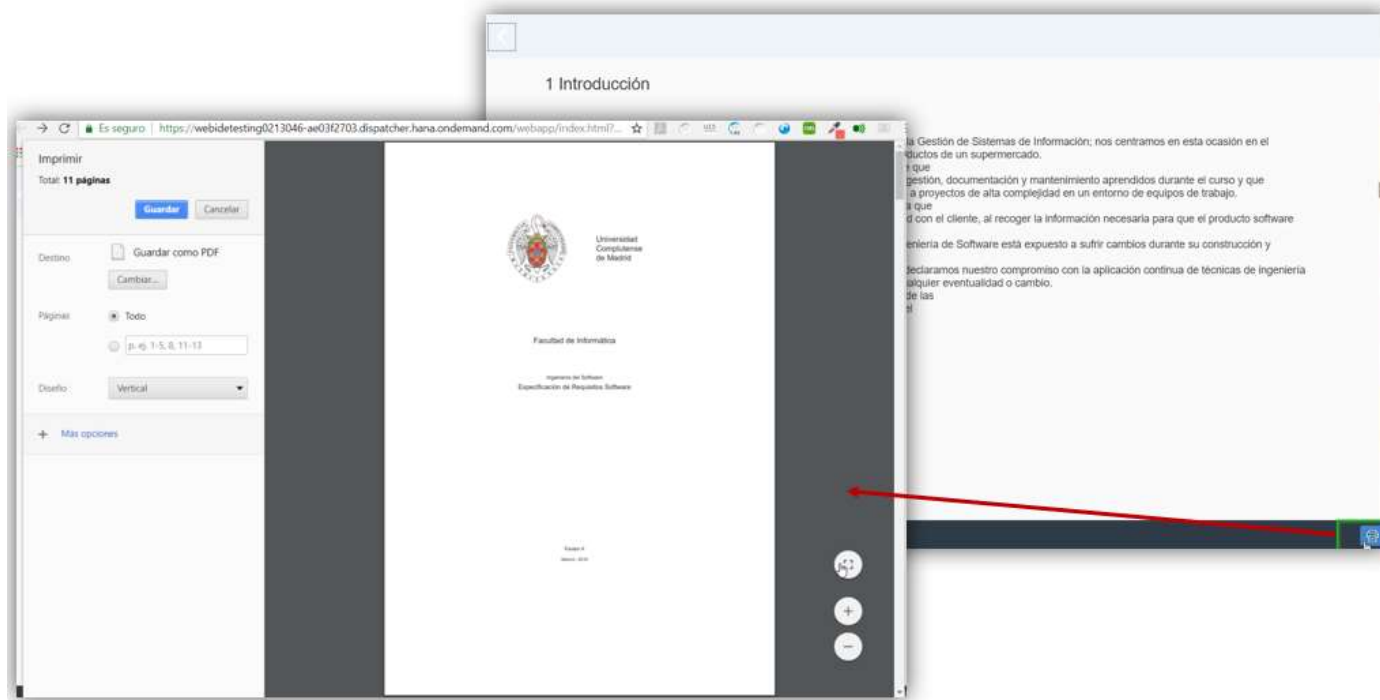
[Figura 39. Implementación – Previsualización de PDF]

- **Impresión del Documento PDF)**

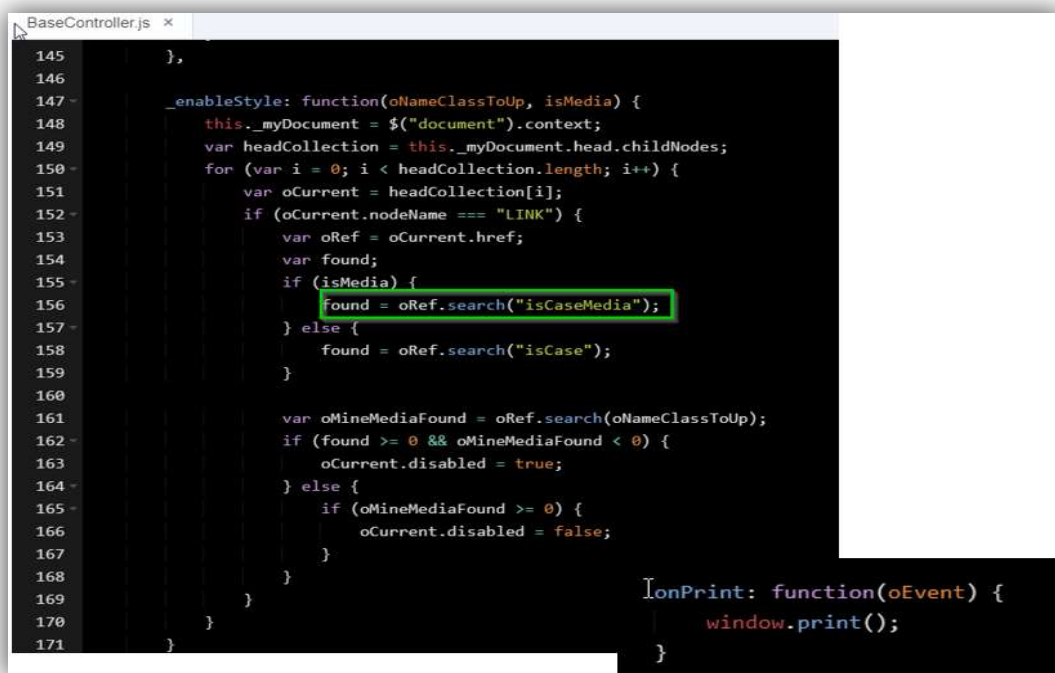
Una vez se accede a la vista de Previsualización PDF, un usuario es capaz de imprimir el documento final SRS.

La acción se realiza al hacer clic en el botón “Imprimir”.

La Herramienta ataca a la Impresión del web browser, aplicando CSS media Queries, para ajustar los parámetros de impresión.



[Figura 40. UI–Impresión PDF]



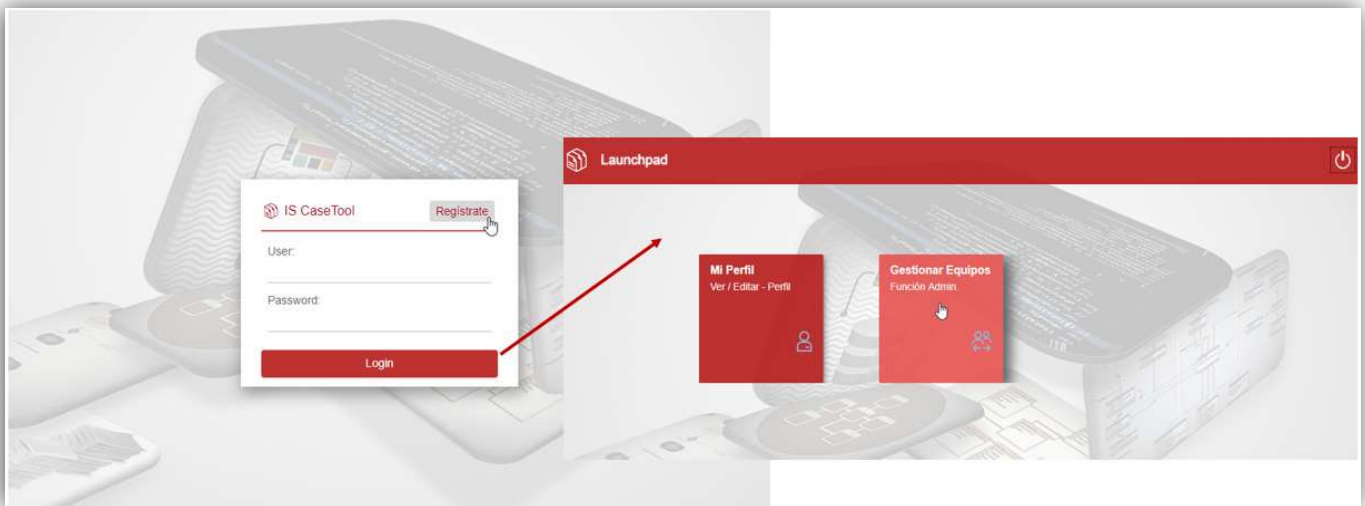
[Figura 41. Implementación–Impresión PDF]

6.2.3.3 UI y Funcionalidades Generales (Todos Los Usuarios)

- **Login)**

Vista principal de la herramienta, establece un punto de acceso común. Presenta un formulario de autenticación que tras ser rellenado por el usuario permite el acceso al catálogo Launchpad tras clic en el botón “Login”.

La herramienta, Comprueba las credenciales contra el back-end, si son válidos guarda los datos de sesión en el web Browser Session y navega al LaunchPad.



[Figura 42. UI – Login]

```

Login.controller.js x
44 - onLogin: function(oEvent) {
45 -     var oView = this.getView();
46 -     oView.setBusy(true);
47 -     var oU = oView.byId("inputU").getValue();
48 -     var oP = btoa(oView.byId("inputP").getValue());
49 -     var oComponent = this.getOwnerComponent();
50 -     var oBundle = this.getResourceBundle();
51 -     var oQuestion = oBundle.getText("NoSessionCorrectComplete");
52 -     var oTitle = oBundle.getText("NoSessionCorrectCompleteTitle");
53 -     var oRouter = this.getRouter();
54 -
55 -     oComponent._initSesion(oU, oP).then(
56 -         function() { // success recived
57 -             var oSession = oComponent.getSession();
58 -             if (oSession.userId) {
59 -                 oView.setBusy(false);
60 -                 oView.byId("inputU").setValue("");
61 -                 oView.byId("inputP").setValue("");
62 -                 oRouter.navTo("launchpad", {
63 -                     userId: encodeURIComponent(oSession.userId),
64 -                     team: encodeURIComponent(oSession.team)
65 -                 }, true);
66 -             } else {
67 -                 oView.setBusy(false);
68 -                 var oConfirmation = {
69 -                     question: oQuestion,
70 -                     title: oTitle
71 -                 };
72 -                 MessageBox.show(oConfirmation.question, {
73 -                     icon: oConfirmation.icon || MessageBox.Icon.WARNING,
74 -                     title: oConfirmation.title,

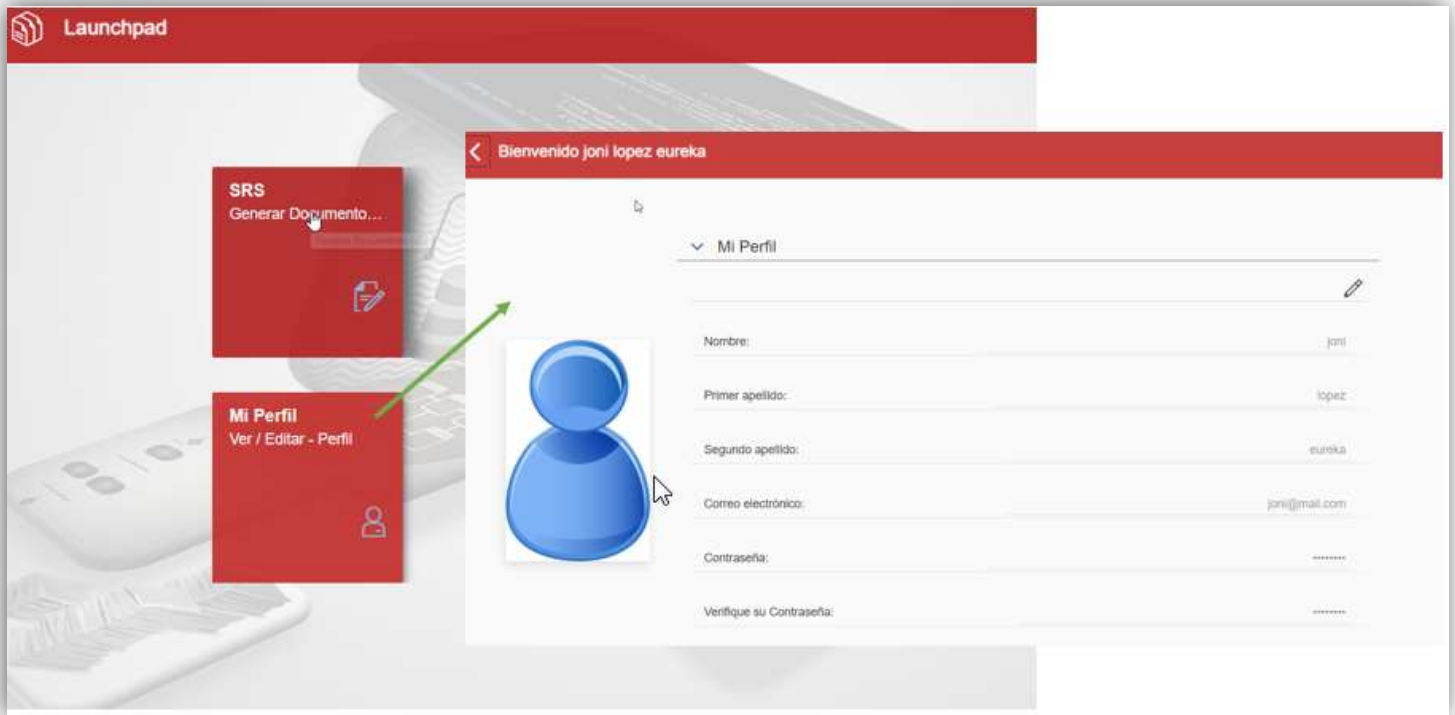
```

[Figura 43. Implementación – Login]

- **Acceso al Perfil de Usuario**

Tras hacer un “LogIn” correcto en la pantalla principal, la herramienta comprueba el tipo de usuario (No Admin) y **navega** al catálogo (Launchpad) donde se muestra el acceso al Perfil de Usuario mediante un Tile.

Tras hacer Clic en el Tile “Mi Perfil”, la herramienta **Navega** a la vista de perfil de usuario.



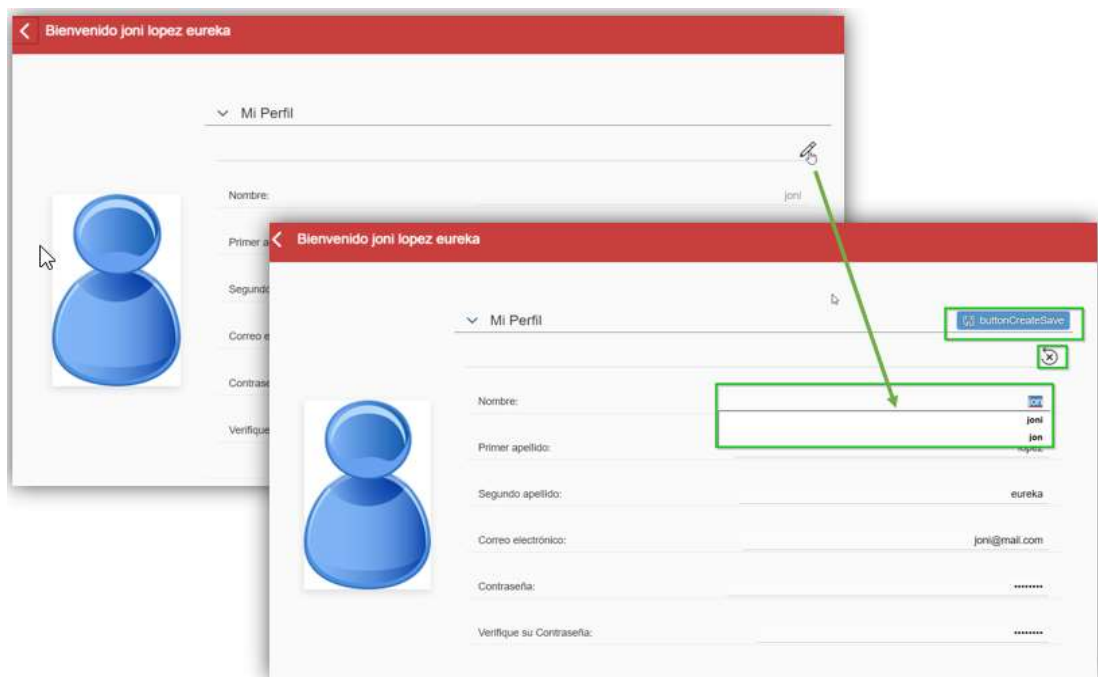
[Figura 44. UI–Acceso a Perfil de Usuario]

- **Modificación de Datos del Usuario**

La vista “Mi Perfil” permite activar la edición de Datos de Usuario (utilizando el botón de Edición) y/o su imagen (haciendo clic sobre la imagen actual).

La herramienta, comprueba la consistencia de datos y activa el botón de guardado.

Tras clic en el botón de “Guardar”, la herramienta envía los datos actualizados al Back-end para persistencia y actualiza la Vista.



[Figura 45. UI-Editar Perfil de Usuario]

```

Profile.controller.js x
366 _updateUser: function(oData) {
367     var that = this;
368     var oFinalData = {
369         "mail": oData.oMailValue,
370         "pass": oData.oPassValue,
371         "name": oData.oNameValue,
372         "lastName": oData.oLastnameValue
373     };
374
375     var oTestedData;
376     if (this._passTouched) {
377         oTestedData = oFinalData;
378     } else {
379         oTestedData = {
380             "mail": oData.oMailValue,
381             "name": oData.oNameValue,
382             "lastName": oData.oLastnameValue
383         };
384     }
385
386     var oModel = this.oDataModel;
387     var oPath = this.getView().getBindingContext().getPath();
388     oModel.update(oPath, oTestedData, {
389         success: function(oDataRes) {
390             MessageToast.show("Todos los Datos se han Guardado Correctamente");
391             that.onEdit(null, that.getView().byId("basicEditButt"), true);
392             that.getView().getElementBinding().refresh(true);
393         },
394         error: function(oError) {
395             MessageToast.show("Error al Guardar los Datos");
396         }
397     });

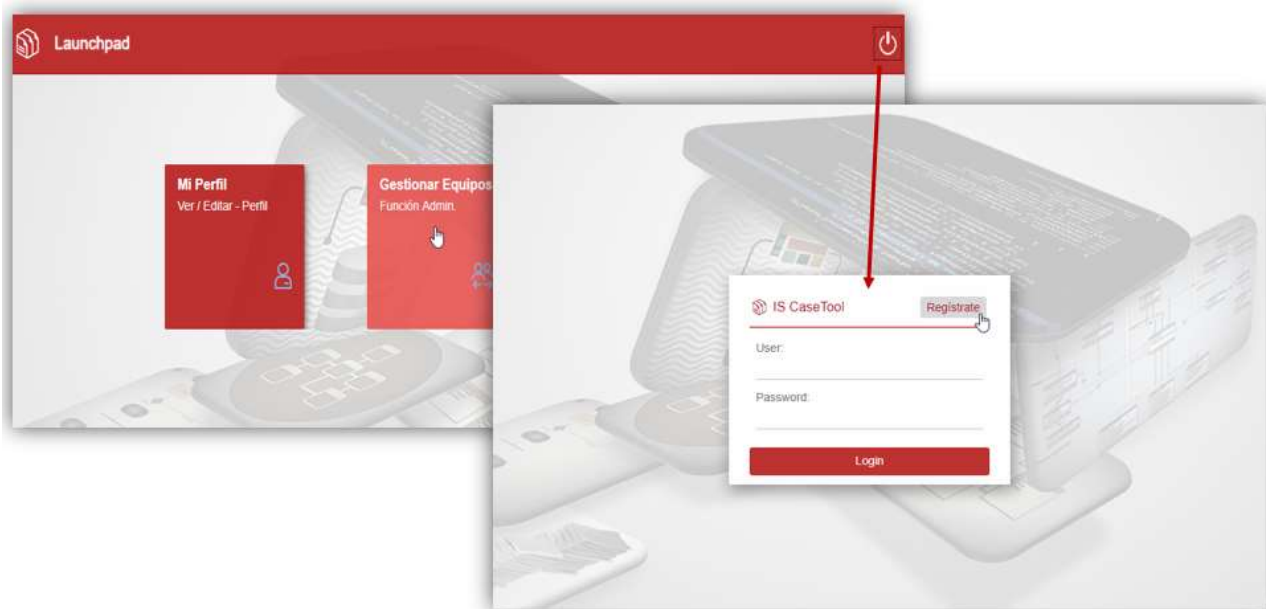
```

[Figura 46. Implementación –Editar Perfil de Usuario]

- **LogOut**

Todos los usuarios pueden cerrar su sesión utilizando el Botón “LogOut” ubicado en el catálogo LaunchPad o bien simplemente cerrando el navegador.

El sistema limpia las variables de Sesión del Browser navega a la página principal “Login”



[Figura 47. UI – LogOut]

```
onLogOut: function(oEvent) {  
    var oComponet = this.getOwnerComponent();  
    oComponet._clearSession();  
    var oRouter = this.getRouter();  
    oRouter.navTo("login", null, true);  
},  
  
onExit: function(oEvent) {  
    this.onLogOut(oEvent);  
}  
  
});
```

[Figura 48. Implementación – LogOut]

7. Conclusiones y Trabajos de Futuro

7.1 Conclusiones

- El trabajo realizado muestra el desarrollo de un proyecto web casi totalmente desarrollado en la parte Front-End aprovechando la tecnología SAP UI 5.
- A pesar de que, de momento, la herramienta **IS Case Tool – Docs** es sólo un prototipo para generar Documentos (actualmente sólo SRS) es suficiente para servir de apoyo a la enseñanza y Aprendizaje de conceptos de Ingeniería del Software.
- **IS Case Tool – Docs**, plantea sin duda una herramienta de acceso web fomentando el desarrollo colaborativo entre quienes la puedan usar.
- El trabajo realizado, demuestra que es completamente loable desarrollar una Aplicación Web – SAPUI 5 atacando un back-end que nada tiene que ver con SAP. Por Tanto, se demuestra a la vez, que es posible **integrar**, con este tipo de aplicaciones y mediante el uso del Protocolo oData, sistemas de distinta índole; se trate de sistemas SAP, externos a SAP o ambos.
- SAP UI5 es, desde mi punto de vista, uno de los Frameworks Web que cuenta con una estructura sólida, librerías basadas en patrones de diseño y facilidad de aprendizaje y uso ágil.
- Finalmente, quiero expresar que todo el esfuerzo realizado durante el desarrollo de este trabajo ha servido para mi crecimiento a nivel profesional e incluso laboral. Las tecnologías aplicadas y la investigación realizada han permitido un continuo aprendizaje.

7.2 Trabajos de Futuro

Como ya se ha comentado con anterioridad, **IS Case Tool – Docs** es una herramienta que se plantea como prototipo en este trabajo. Por tanto, quedan muchas tareas abiertas y por resolver, entre ellas:

- **Autenticación back-end Basada en Token**, según la documentación investigada sobre n-oData-server (API Rest) el proyecto está basado en el conocido framework LoopBack y por tanto existe la posibilidad de extender el back-end, incluyendo una mejor autenticación y seguridad.
- **Plantear otro tipo de back-end**, sabiendo que el front-end está preparado para atacar cualquier servicio que maneje oData. Cabe la posibilidad de implementar un back-end más completo, sobre el cual el programador tenga un mejor control. Según algunas investigaciones realizadas, por ejemplo, es lícito desarrollar un servicio oData API REST desde cero utilizando Visual Basic y librerías Microsoft sobre Distintas Bases de Datos. Esto constituye un punto de Investigación y Desarrollo interesante para el futuro.
- **Completar desarrollo y Tratamiento de Plantillas**, puesto que actualmente solo se puede Generar un SRS, la funcionalidad de la herramienta queda limitada, aunque su estructura haya sido preparada para generar más documentos (P.e. Plan de Proyecto y Memoria Final de IS).
- **Directores y Editores**, son campos que se plantean en el modelo de datos que hacen referencia al director de un equipo de trabajo y al usuario que puede editar un contenido. Esto se planteó en principio con el fin de añadir una entrada en el Launchpad que permita Gestionar un Equipo

de manera interna. Es decir, designando un director que fuese capaz de gestionar las tareas o apartados que deben rellenar los miembros de su equipo.

- **Mejoras**, a realizar sobre el CSS de Media Queries desarrollado para la impresión nativa. O incluso agregar una librería externa JS que se encargue de generar PDF.

7. Conclusions and Future Works

7.1 Conclusions

- The work carried out shows the development of a web project almost totally developed in the Front-End part, taking advantage of SAP UI 5 technology.
- Although, at the moment, the IS Case Tool - Docs is only a prototype to generate Documents (currently only SRS) is sufficient to support the teaching and Learning of Software Engineering concepts.
- **IS Case Tool - Docs**, undoubtedly proposes a web access tool promoting collaborative development among those who can use it.
- The work done shows that it is completely doable to develop a Web Application - SAPUI 5 attacking a back-end that has nothing to do with SAP. Therefore, it is demonstrated at the same time, that it is possible to integrate, with this type of applications and through the use of the Protocol oData, systems of different nature, whether SAP systems, external to SAP or both.
- SAP UI5, is from my point of view, one of the Web Frameworks that has a solid structure, libraries based on design patterns and ease of learning and agile use.
- Finally, to express that all the effort made during the development of this work, has served for my growth to professional and even laboral level. The applied technologies and the research carried out have allowed a continuous learning.

7.2 Future Works

As previously mentioned, IS Case Tool - Docs is a tool that is proposed as a prototype in this work. There are therefore many open and unresolved tasks, among them:

- **Back-end authentication Token-based**, according to the documentation researched on n-oData-server (API Rest) the project is based on the well-known LoopBack framework and therefore there is the possibility of extending the back-end, including better authentication and security.
- **Raise another type of back-end**, knowing that the front-end is ready to attach any service that handles oData. It is possible to implement a more complete back-end, over which the

programmer has a better control, according to some investigations, for example; it is licit to develop an oData API REST service, from scratch using Visual Basic and Microsoft libraries on Different Databases. This constitutes an interesting Research and Development point for the future.

- **Complete development and Treatment of Templates**, since currently only one SRS can be generated, the functionality of the tool is limited, even though its structure has been prepared to generate more documents (e.g. Project Plan and IS Final Report).
- **Directors and Editors**, are fields that are raised in the data model that refer a work team's director and the user who can edit a content. This was initially proposed in order to add an entry in the Launchpad that allows you to Manage a Team internally. That is to say, designating a director who would be able to manage the tasks or sections that the members of his team must fill in.
- **Improvements**, to be made on the CSS of Media Queries developed for native printing. Or even add an external JS library to generate PDF.

8. Contribuciones

El desarrollo de este trabajo se ha realizado de manera individual. Por tanto, las únicas contribuciones que cabe mencionar son las propias del autor y aquellas directrices y enseñanzas del Director de Proyecto, Antonio Sarasa.

Bibliografía

- SAP [online], disponible en: <https://sapui5.hana.ondemand.com> [1].
- SAP UI5 - SDK [online], disponible en: <https://sapui5.hana.ondemand.com> [2].
- FIORI [online], disponible en: <https://www.sap.com/spain/products/fiori.html> [3].
- HTML5 [online], disponible en: <https://www.w3schools.com/html> [4].
- JavaScript [online], disponible en: <https://www.w3schools.com/js> [5].
- oData [online], disponible en: <https://www.odata.org> [6].
- CSS3- SDK [online], disponible en: <https://www.w3schools.com/css> [7].
- JSON [online], disponible en: https://www.w3schools.com/js/js_json_intro.asp [8].
- HTTP-Verbos [online], disponible en:
<https://developer.mozilla.org/es/docs/Web/HTTP/Methods> [9].
- Npm n-odata-server, disponible en:
<https://www.npmjs.com/package/n-odata-server> [10].
- n-odata-server [online], disponible en:
<https://github.com/htammen/n-odata-server/wiki> [11].
- LoopBack – Framework [online], disponible en: <https://loopback.io/> [12].
- Mongo DB [online], disponible en: <https://www.mongodb.com> [13].
- Apache – Server [online], disponible en: <https://www.apache.org> [14].
- XAMPP [online], disponible en: <https://www.apachefriends.org/es/index.html> [15].
- SAP Web IDE [online], disponible en:
<https://www.sap.com/spain/developer/topics/sap-webide.html> [16].
- SAP Cloud Platform [online], disponible en:
<https://www.sap.com/developer/topics/cloud-platform.h>
https://es.wikipedia.org/wiki/DNS_din%C3%A1micotml?trial=%2Fregistration%2Ftrial.908cb719-0e03-421c-a091-daca045f0acc.html [17].
- GitHub [online], disponible en: <https://github.com> [18].
- DynDNS [online], disponible en:
https://es.wikipedia.org/wiki/DNS_din%C3%A1mico [19].

- **Cliente-Servidor [online]**, disponible en: <https://es.wikipedia.org/wiki/Cliente-servidor> [20].
- **Guía de Diseño Fiori [online]**, disponible en: <https://experience.sap.com/fiori-design-web/get-started/> [21].
- **Conceptos Trabajo en Equipo [online]**, disponible en: https://es.wikipedia.org/wiki/Trabajo_en_equipo [22].
- **Google Docs Mission [online]**, disponible en: <https://www.google.com/intl/es-419/about/> [23].
- **Google Docs [online]**, disponible en: <https://www.google.es/intl/es/docs/about/> [24].
- **BPNM [online]**, disponible en: <http://www.bpmn.org/> [25].
- **Heflo [online]**, disponible en: <https://www.heflo.com/es/herramienta-de-documentacion-de-procesos-de-negocio/> [26].
- **REM [online]**, disponible en: http://www.lsi.us.es/descargas/descarga_programas.php?id=3 [27].
- **Enterprise Architect [online]**, disponible en: <http://www.sparxsystems.com.au/products/ea/index.html> [28].
- **IberDok [online]**, disponible en: <http://iberdok.com/#caracteristicas> [29].
- **OpenKm [online]**, disponible en: <https://www.openkm.com/es/#GestionDocumental> [30].

Anexos

Apéndice A. Guía de Instalación de la Herramienta.

La herramienta se entrega, en un fichero “**iscasetool.Zip**” que contiene un directorio Raíz principal “**IScasetool**” dentro del cual se separan dos directorios principales “**back-end**” y “**front-end**”. Que llevan toda la implementación de sus respectivas partes.

Cada parte se puede Montar en una máquina Diferente, siempre y cuando se configuren correctamente las conexiones.

Para instalar la herramienta se deben seguir los siguientes pasos:

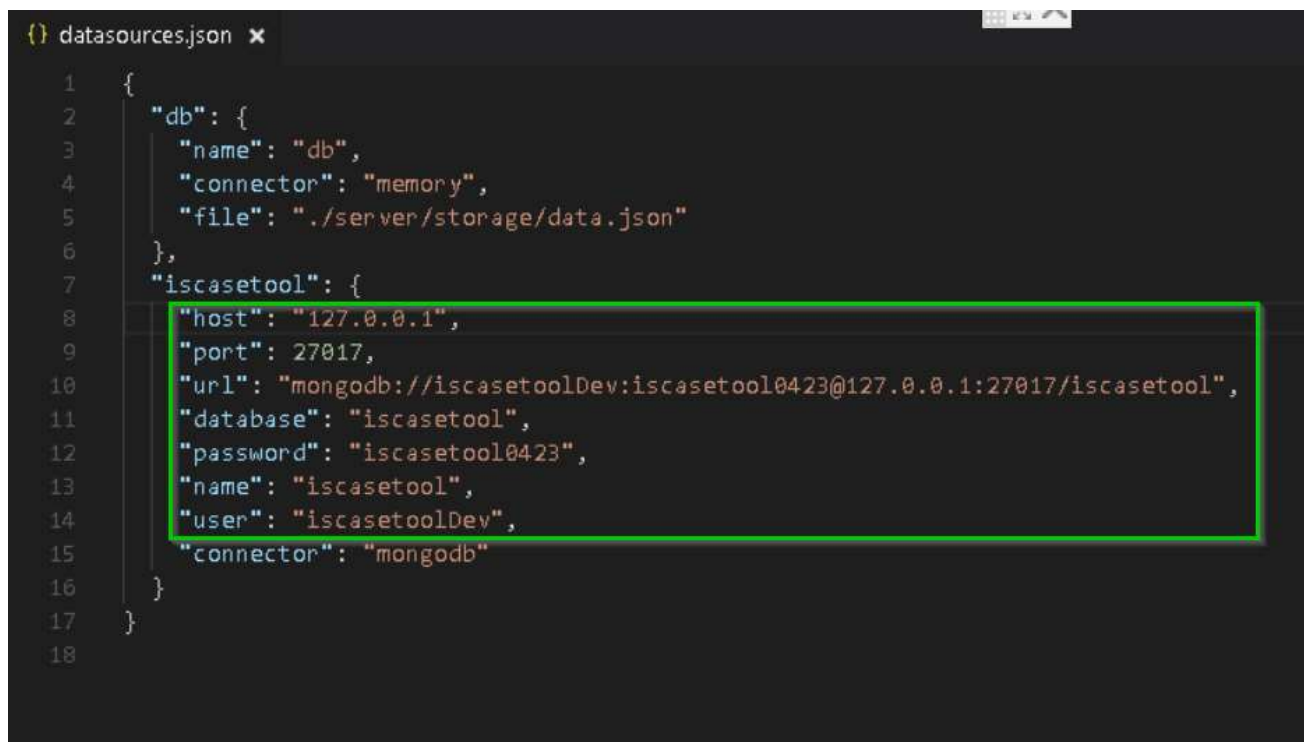
a) Back-end:

- Instalar **Node JS** en la máquina en la que correrá el back-end. Para Ello ir a <https://nodejs.org/en/download/> y escoger una de las descargas o instalaciones apropiada para la máquina objetivo.
Seguir las instrucciones de instalación, indicadas en la web.
- Es requisito fundamental disponer **en la máquina back-end** de una base de datos **MONGO DB**, o bien de alguna conexión distribuida en Cloud. Más información en: <https://www.mongodb.com/>.

El back-end, **n-odata-server**, ataca dicha base datos mediante la configuración del fichero “**IScasetool/back-end/server/datasources.json**” que debe editarse con los parámetros de conexión a la base de datos que se va a atacar.

La siguiente figura Muestra, la configuración actual del fichero, que no es otra, sino la que se ha dado para la entrega productiva.

Los campos que se deben editar son los contenidos en el recuadro Verde:



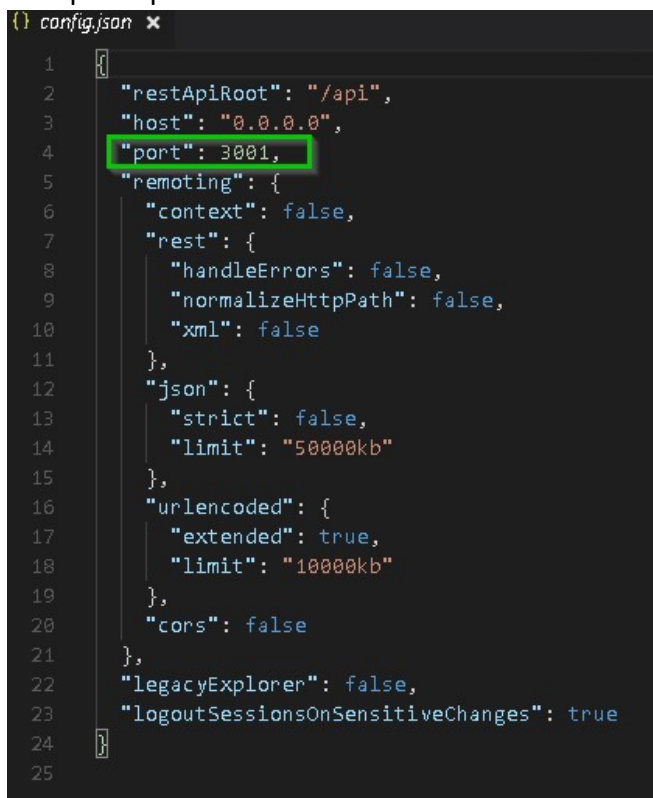
```

1  {
2    "db": {
3      "name": "db",
4      "connector": "memory",
5      "file": "../server/storage/data.json"
6    },
7    "iscasetool": {
8      "host": "127.0.0.1",
9      "port": 27017,
10     "url": "mongodb://iscasetoolDev:iscasetool0423@127.0.0.1:27017/iscasetool",
11     "database": "iscasetool",
12     "password": "iscasetool0423",
13     "name": "iscasetool",
14     "user": "iscasetoolDev",
15     "connector": "mongodb"
16   }
17 }
18

```

[Figura 49. Configuración Mongo sobre n-odata-server]

- El servidor se montará como Servicio API REST, por defecto sobre el puerto 3001 si se desea se puede cambiar dicho puerto en el fichero "**IScasetool/back-end/server/config.json**". Siempre y cuando se sepa lo que se está haciendo.



```

1  {
2    "restApiRoot": "/api",
3    "host": "0.0.0.0",
4    "port": 3001,
5    "remoting": {
6      "context": false,
7      "rest": {
8        "handleErrors": false,
9        "normalizeHttpPath": false,
10       "xml": false
11     },
12     "json": {
13       "strict": false,
14       "limit": "50000kb"
15     },
16     "urlencoded": {
17       "extended": true,
18       "limit": "10000kb"
19     },
20     "cors": false
21   },
22   "legacyExplorer": false,
23   "logoutSessionsOnSensitiveChanges": true
24 }
25

```

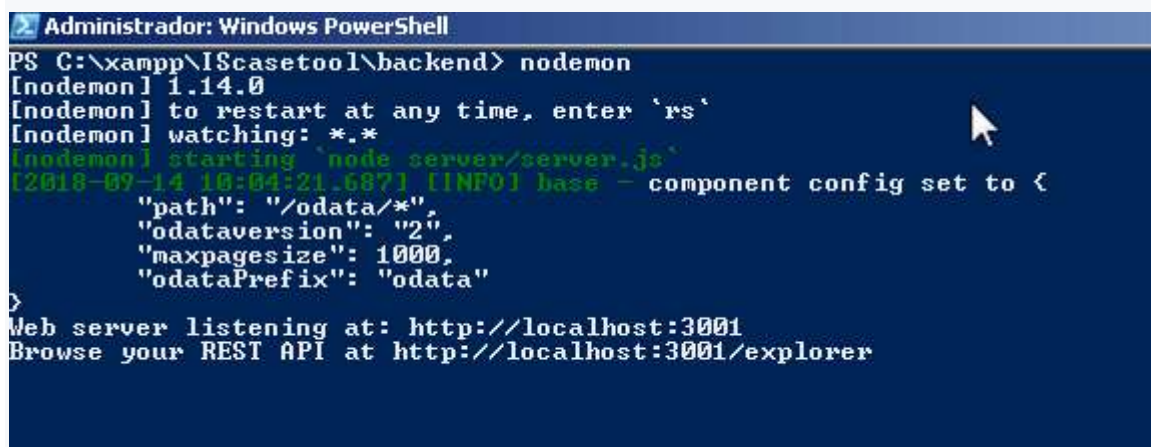
[Figura 50. Configuración del puerto Escucha sobre n-odata-server API REST]

- **Instalar LoopBack**, dado que n-odata-server se basa en el conocido framework “LoopBack” se debe instalar éste previamente. Para ello, abrir una terminal de línea de comandos y escribir:
npm install -g strongloop

Con este paso, sería suficiente para empezar, puesto que el resto de configuración de **n-odata-server** ya se entrega preparado.

Para más información sobre la instalación de **n-odata-server**; visitar : https://github.com/htammen/n-odata-server/wiki/tutorial_01#install-loopback

- **Finalmente**, ha de arrancarse el servidor:
 - Para ello abrir una terminal y dirigirse hasta el directorio “back-end”
 - Una vez se encuentre en la ruta ejecutar:
npm install -g nodemon instala el demonio de auto recarga de código
 - Tras terminar la instalación anterior ejecutar el comando: **nodemon**
 - El servidor deberá, entonces empezar a funcionar y se mostrará por la línea de comandos algo similar a lo siguiente:



```

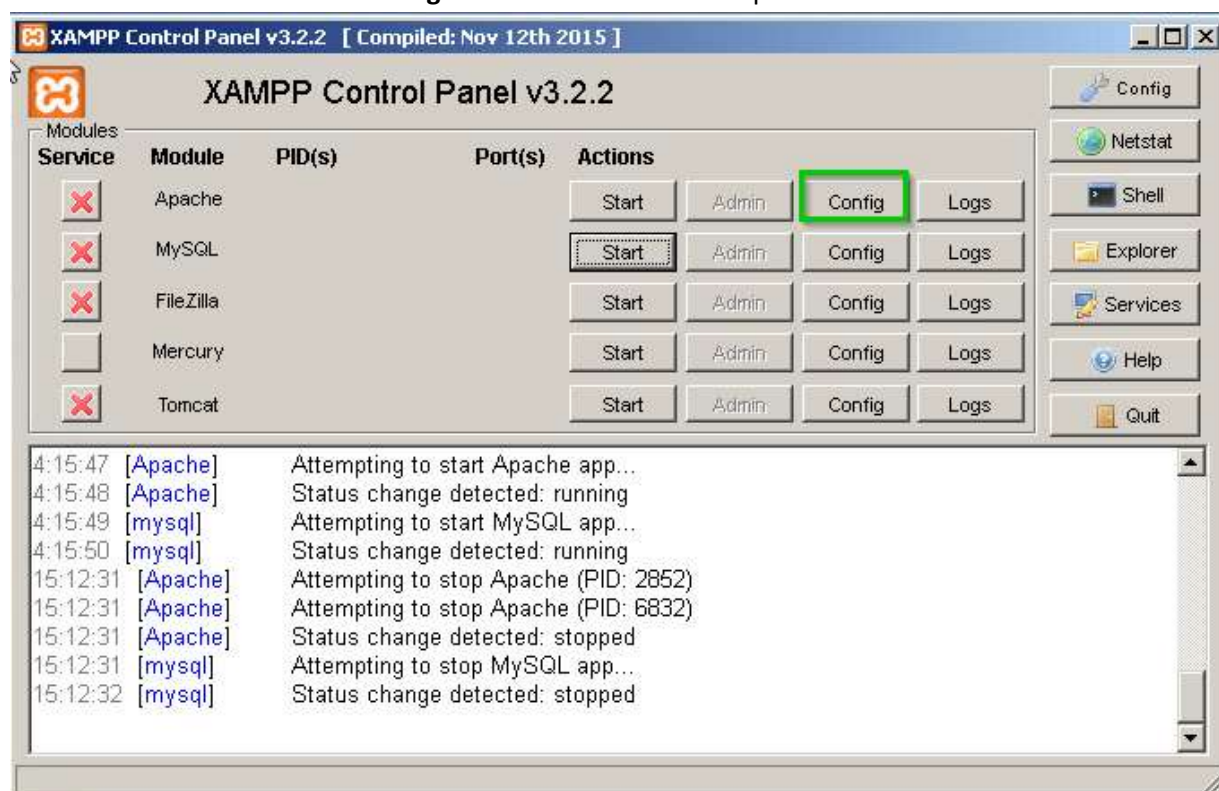
Administrador: Windows PowerShell
PS C:\xampp\IScasetool\backend> nodemon
[nodemon] 1.14.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node server/server.js`
[2018-09-14 10:04:21.687] [INFO] base - component config set to {
  "path": "/odata/*",
  "odataversion": "2",
  "maxpagesize": 1000,
  "odataPrefix": "odata"
}
Web server listening at: http://localhost:3001
Browse your REST API at http://localhost:3001/explorer
  
```

[Figura 51. Arranque correcto de oData API REST]

- **Es recomendable**, además, disponer de un DNS o DynDNS que reference a la máquina back-end desde el exterior traduciendo la IP. De lo contrario las configuraciones aquí mostradas tendrán que hacerse a través de la IP pública.
En el caso de la Demo entregada en producción, para este trabajo tenemos como punto de acceso a la API la url:
<http://iscasetool.hopto.org:3001/explorer/>
que nos da acceso a la interfaz de exploración de la propia API.

b) front-End:

- **Levantar un servidor APACHE**, para servir las páginas front-end. Se puede usar cualquier distribución APACHE cuyo directorio principal a servir sea **"IScasetool/frontend/"**. En esta guía se usa, para ello la distribución XAMPP.
- **Instalar XAMPP**, en la máquina que servirá las páginas de front-end. Para saber más, puede visitar: <https://www.apachefriends.org/es/index.html> **descargar la versión adecuada y seguir las guías de instalación.**
- **Cambiar el directorio Raíz de servicio en XAMPP** para el servidor Apache, para ello:
 - Acceder a la interfaz gráfica de XAMPP
 - Clic en el botón **"Config"** de la línea del servidor Apache



[Figura 52. Configuración de XAMPP]

- Abrir el fichero de configuración **"httpd.conf"**
- Buscar la directiva **"DocumentRoot"** dentro del fichero. Debe quedar algo similar a la siguiente imagen, de acuerdo a donde se tenga almacenado el proyecto. Se recomienda mantener la estructura de la imagen.

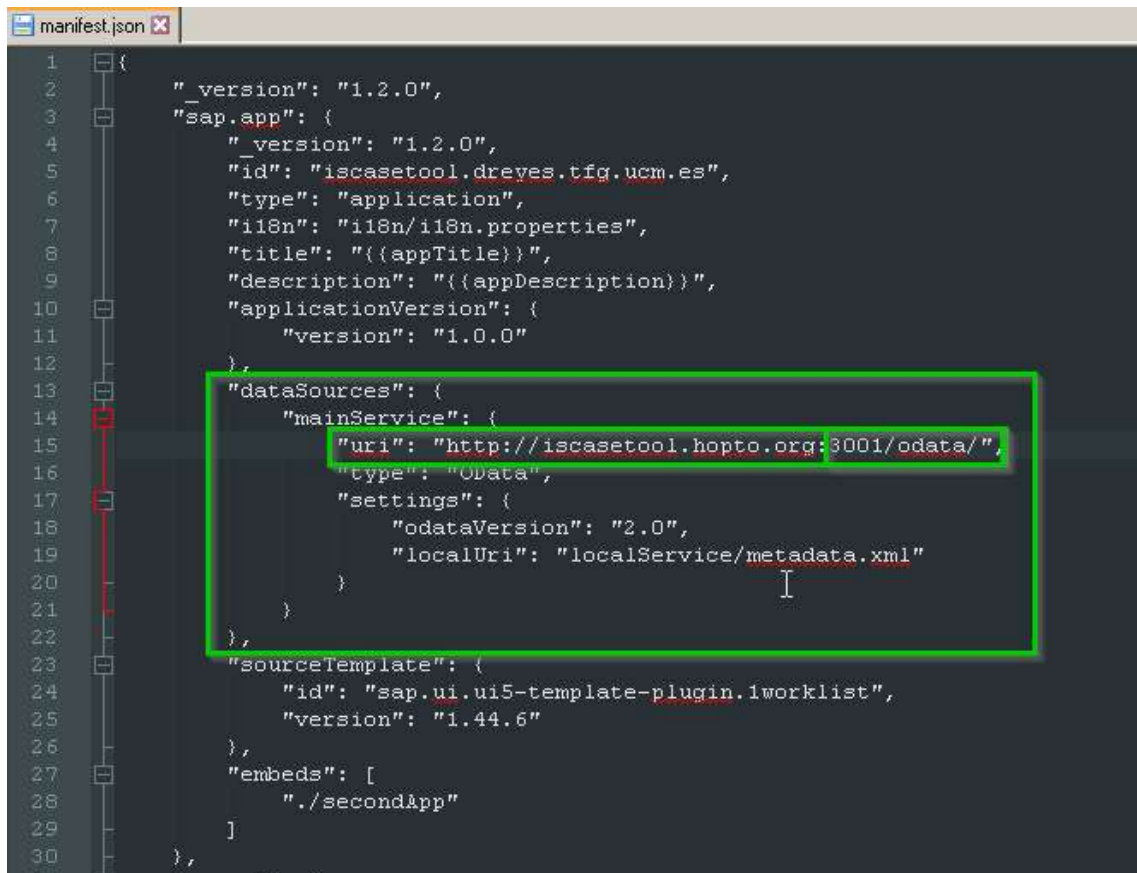
```

247 DocumentRoot "C:/xampp/IScasetool/frontend"
248 <Directory "C:/xampp/IScasetool/frontend">
  
```

[Figura 53. Configuración de Directorio en XAMPP]

- Se recomienda además cambiar el puerto escucha del servidor apache por uno diferente al 80. Puesto que es posible que se encuentre ocupado. Más información en <https://www.srcofigofuente.es/tutoriales/ver-tutorial/cambiar-puerto-apache-xampp>

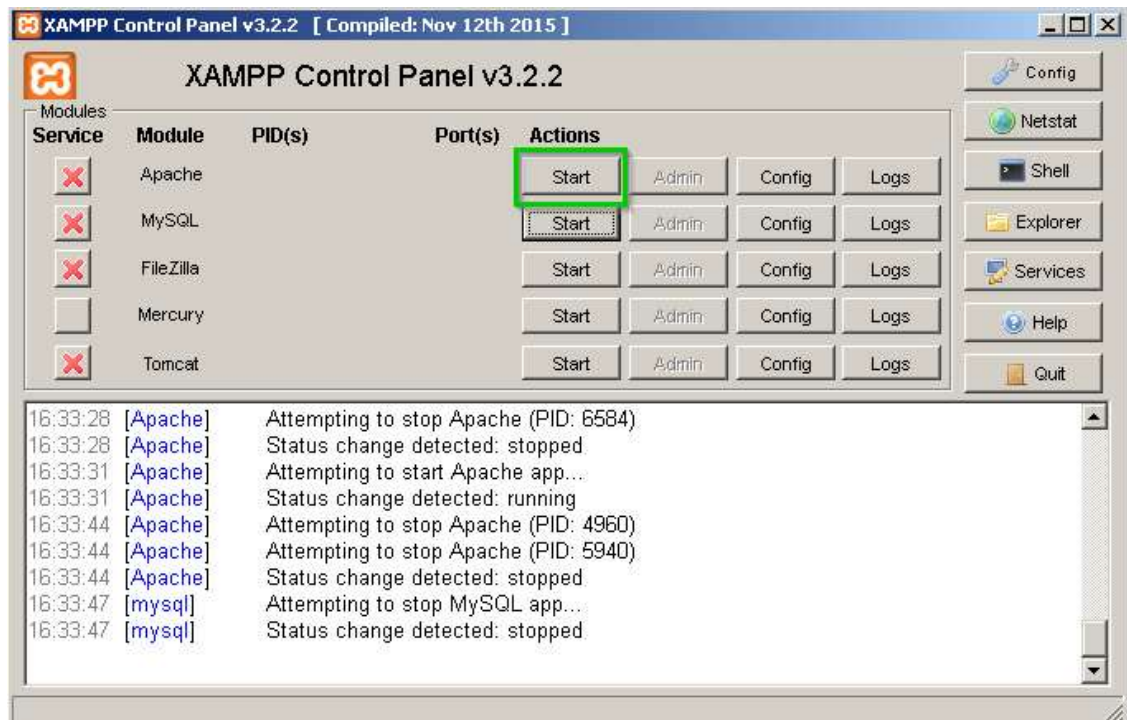
- **Configurar el servicio oData como recurso de datos en el front-end. Para ello:**
 - Abrir el fichero “IScasetool/front-end/manifest.json”
 - Verificar la URI que referencia al servicio oData configurado en el back-end. El puerto debe coincidir con lo descrito en la [Figura 50.]
- La siguiente figura muestra la configuración por defecto de la URI basada en el servicio presentado en producción sobre este trabajo.



```
1 {
2   "_version": "1.2.0",
3   "sap.app": {
4     "_version": "1.2.0",
5     "id": "iscasetool.dreyes.tfg.ucm.es",
6     "type": "application",
7     "i18n": "i18n/i18n.properties",
8     "title": "{{appTitle}}",
9     "description": "{{appDescription}}",
10    "applicationVersion": {
11      "version": "1.0.0"
12    },
13    "dataSources": {
14      "mainService": {
15        "uri": "http://iscasetool.hopto.org:3001/odata/",
16        "type": "OData",
17        "settings": {
18          "odataVersion": "2.0",
19          "localUri": "localService/metadata.xml"
20        }
21      },
22    },
23    "sourceTemplate": {
24      "id": "sap.ui.ui5-template-plugin.1worklist",
25      "version": "1.44.6"
26    },
27    "embeds": [
28      "./secondApp"
29    ]
30  },
31 }
```

[Figura 54. Configuración de conexión a oData desde Front-end]

- **Finalmente, Iniciar el servidor Apache** desde la interfaz de XAMPP. Al finalizar la herramienta será accesible desde el navegador web con la URL o IP Configurada.



[Figura 55. Iniciar Apache desde XAMPP]

- **Como Recomendación Final**, no se debe olvidar que antes de poner en producción la herramienta y/o publicarla; es conveniente dar de alta a Miembro/s Administrador/es. Activando para ello, el flag dispuesto en el fichero "IScasetool/frontend/config/config.json". Que actúa sobre la UI permitiendo el alta de dichos Miembros. Tras terminar las altas, se debe volver a desactivar.



[Figura 56. Configuración de Administrador Front-end]

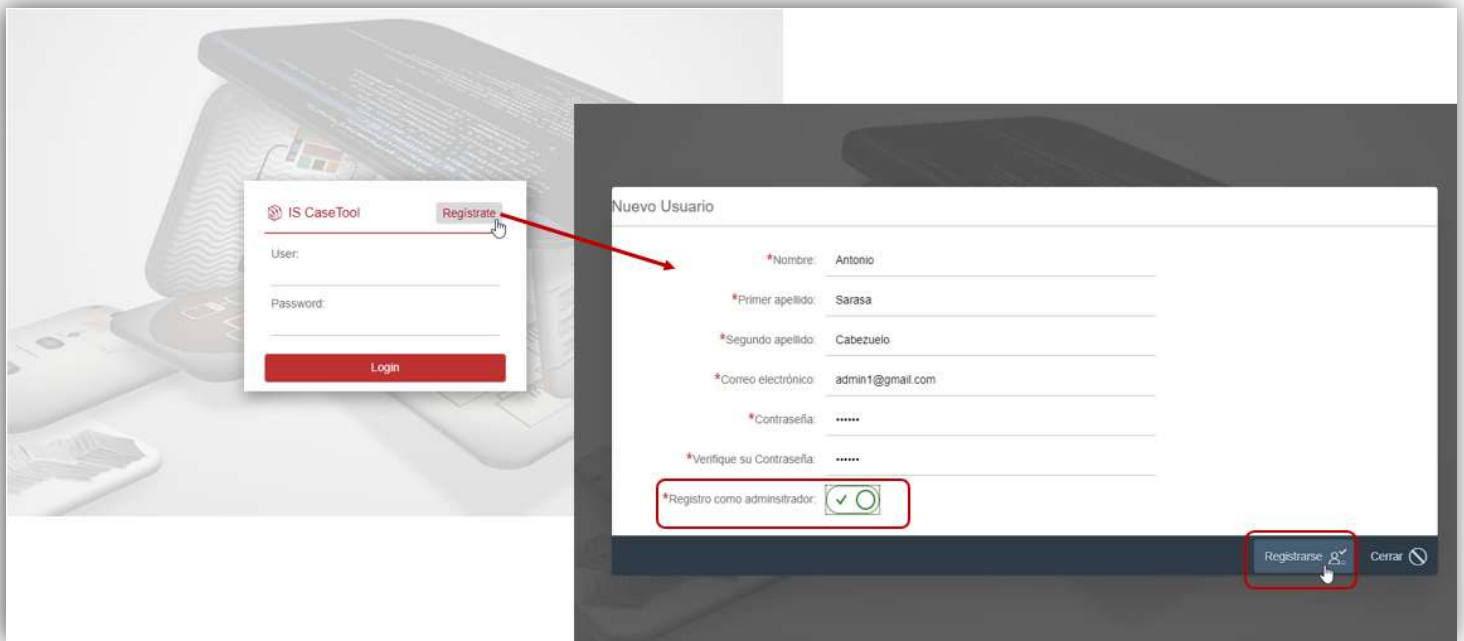
Apéndice B. Manual de Usuario.

A continuación, se explica cómo utilizar las principales funcionalidades ofrecidas por “IS Case Tool – Docs”, desde un punto de vista enfocado a los usuarios.

Para ello sólo hace falta seguir los pasos descritos a continuación:

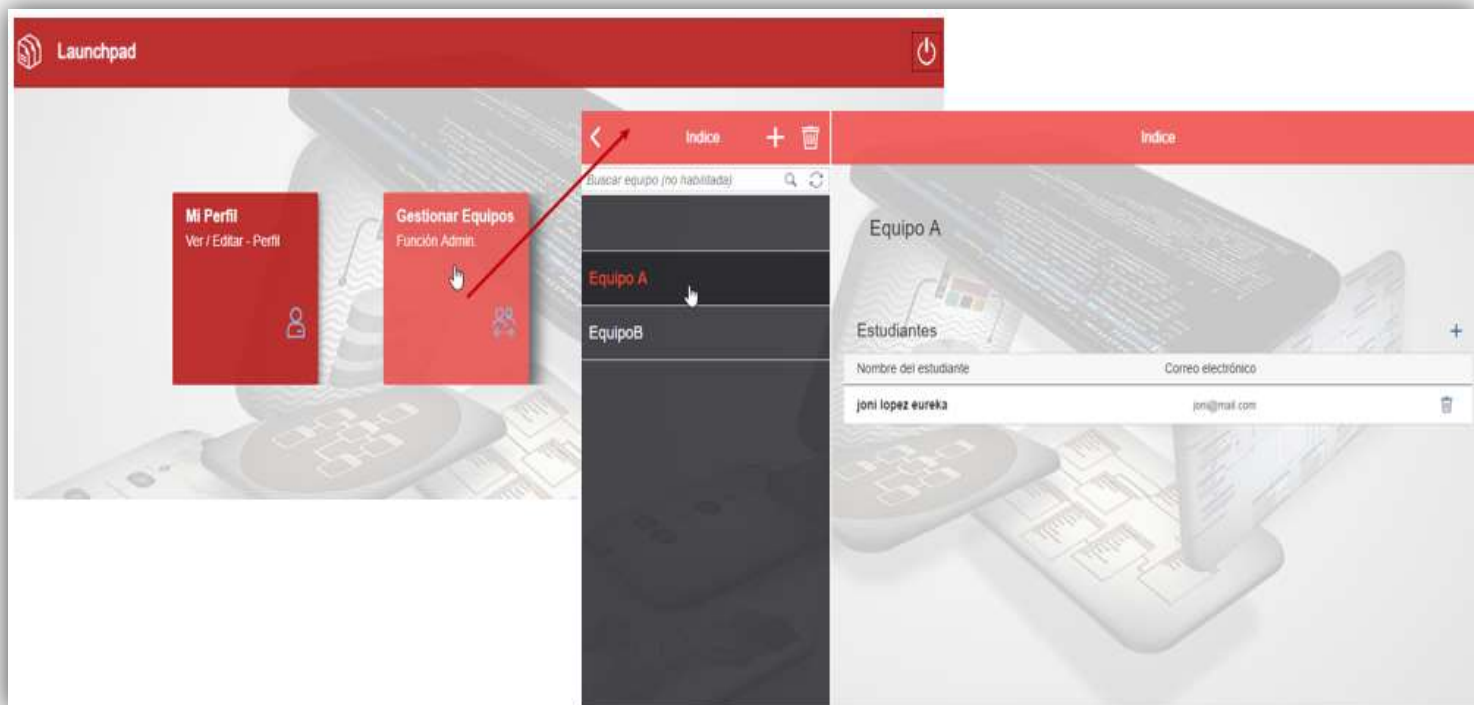
Para Miembro Administrador:

- **Registro de Usuario)**
 - Acceder a la herramienta en la vista principal “Login”
 - Hacer clic en el botón “**Regístrate**”, se abrirá un formulario a rellenar.
 - Rellenar los campos solicitados de forma coherente. De lo contrario no es posible realizar el Registro.
 - Activar el botón “Switch” de “**Registro como Administrador**”
 - Se activa el botón “**Registrarse**” al pie del formulario.
 - Clic en el botón “**Registrarse**”. Una vez creado el Usuario en la BD, se muestra un mensaje de confirmación.



[Figura 57. Réplica de Figura 17.]

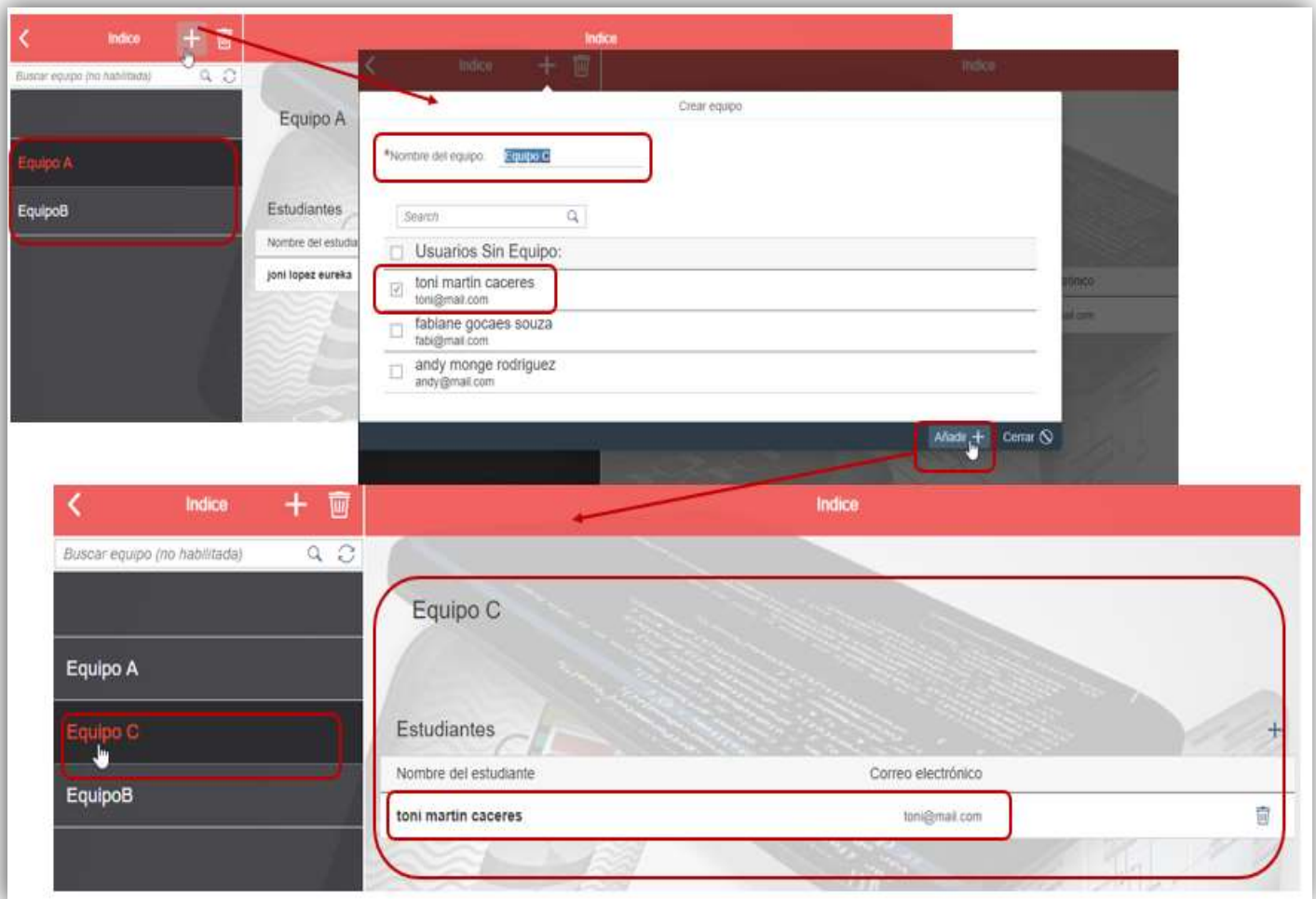
- **Acceso a Gestión de Equipos)**
 - Realizar un **“Login”** siendo Miembro Administrador (ver Funcionalidades Generales)
 - Desde el catálogo **“Launchpad”** clic sobre el Tile **“Gestión de Equipos”**.



[Figura 58. Réplica de Figura 19.]

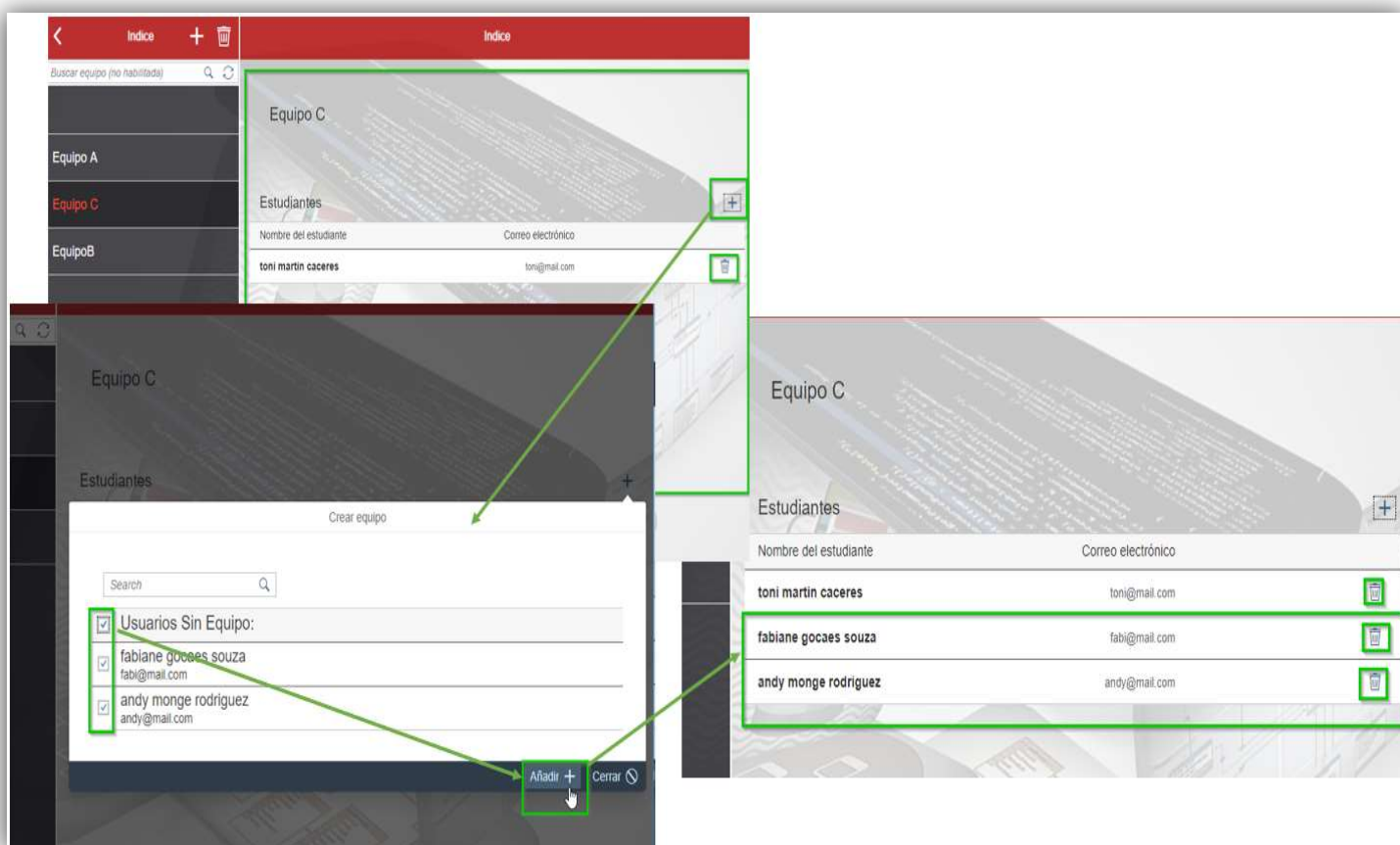
- **Alta de un Equipo)**

- Desde la Vista de “**Gestión de Equipos**”, clic sobre el botón “**Agregar Equipo**” en la cabecera del menú de la izquierda
- Se abre un **formulario y una lista de Miembros sin Equipo Asignado**
- Definir el Nombre del Nuevo Equipo
- Seleccionar los Miembros Integrantes (al menos uno) para el nuevo Equipo.
- Se activa el botón “**Añadir**” al pie del Formulario.
- Clic en el botón “**Añadir**”.
- El Nuevo equipo creado aparece en la lista.



[Figura 59. Réplica de Figura 21]

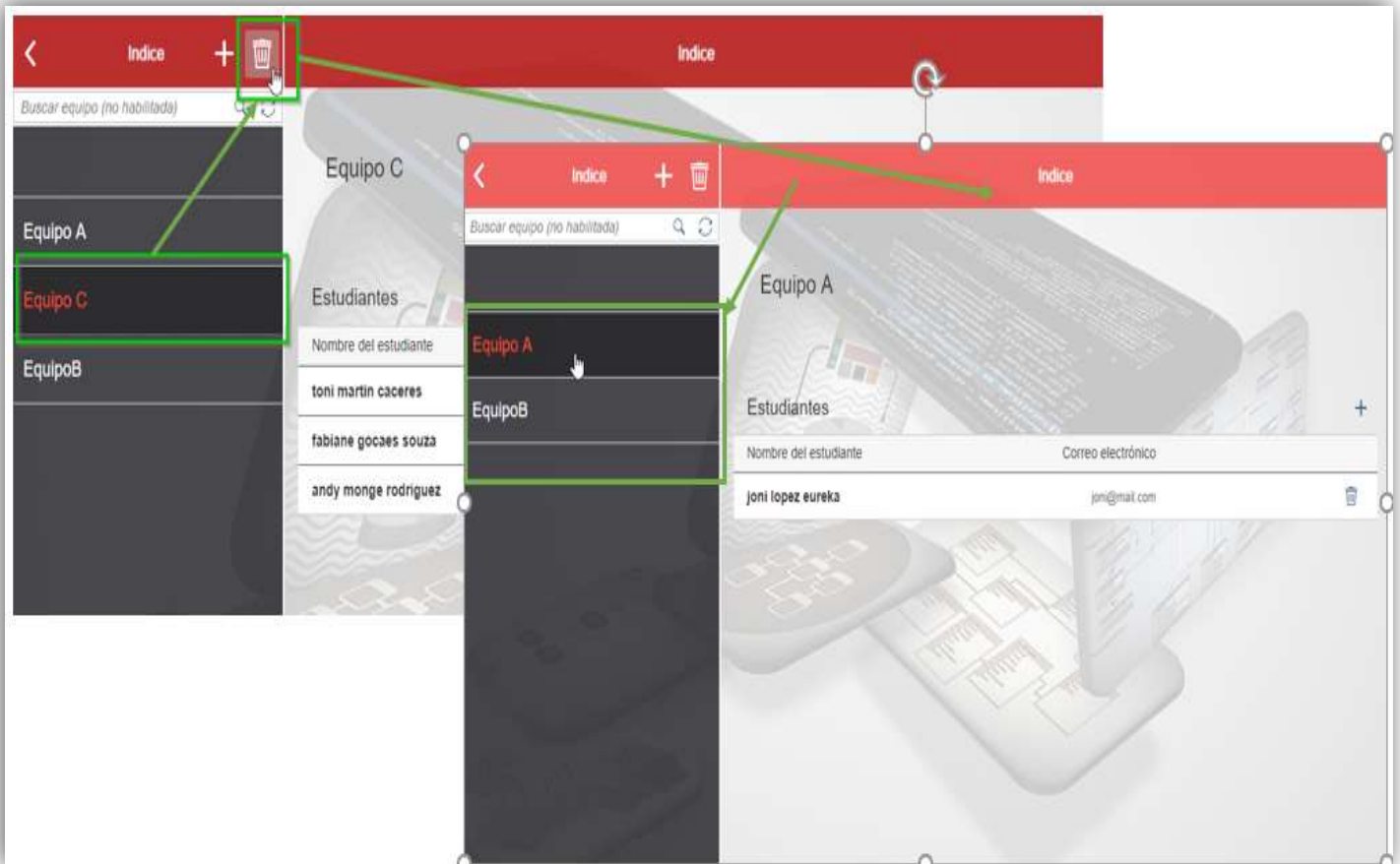
- **Agregar Miembros a un Equipo)**
 - Acceder a la “**Gestión de Equipos**”
 - Seleccionar Equipo concreto en el menú de la izquierda.
 - Clic en el botón “+”(añadir Miembro). Ubicado en la cabecera de **la lista de Miembros** del equipo Seleccionado.
 - Se abre una lista de Miembros disponibles.
 - Seleccionar uno o más usuarios de la lista mostrada
 - Se activa el botón “**Añadir**” al pie de la Lista de Miembros Seleccionados.
 - Clic en el botón “**Añadir**”.
 - Los miembros Nuevos aparecen como parte del Equipo.
- **Eliminar Miembro de un Equipo)**
 - Acceder a la “**Gestión de Equipos**”
 - Seleccionar Equipo concreto en el menú de la izquierda.
 - Clic en el botón “**Eliminar**” mostrado junto a la información del Miembro que se desea excluir del Equipo.
 - El miembro excluido desaparece de la Lista de Miembros del Equipo.



[Figura 60. Réplica de Figura 23.]

- **Baja de un Equipo)**

- Acceder a la **“Gestión de Equipos”**
- Seleccionar Equipo concreto en el menú de la izquierda.
- Clic en el botón **“Eliminar”** mostrado en la **cabecera del Menú de la Izquierda**.
- El Equipo desaparece del Menú de la Izquierda.

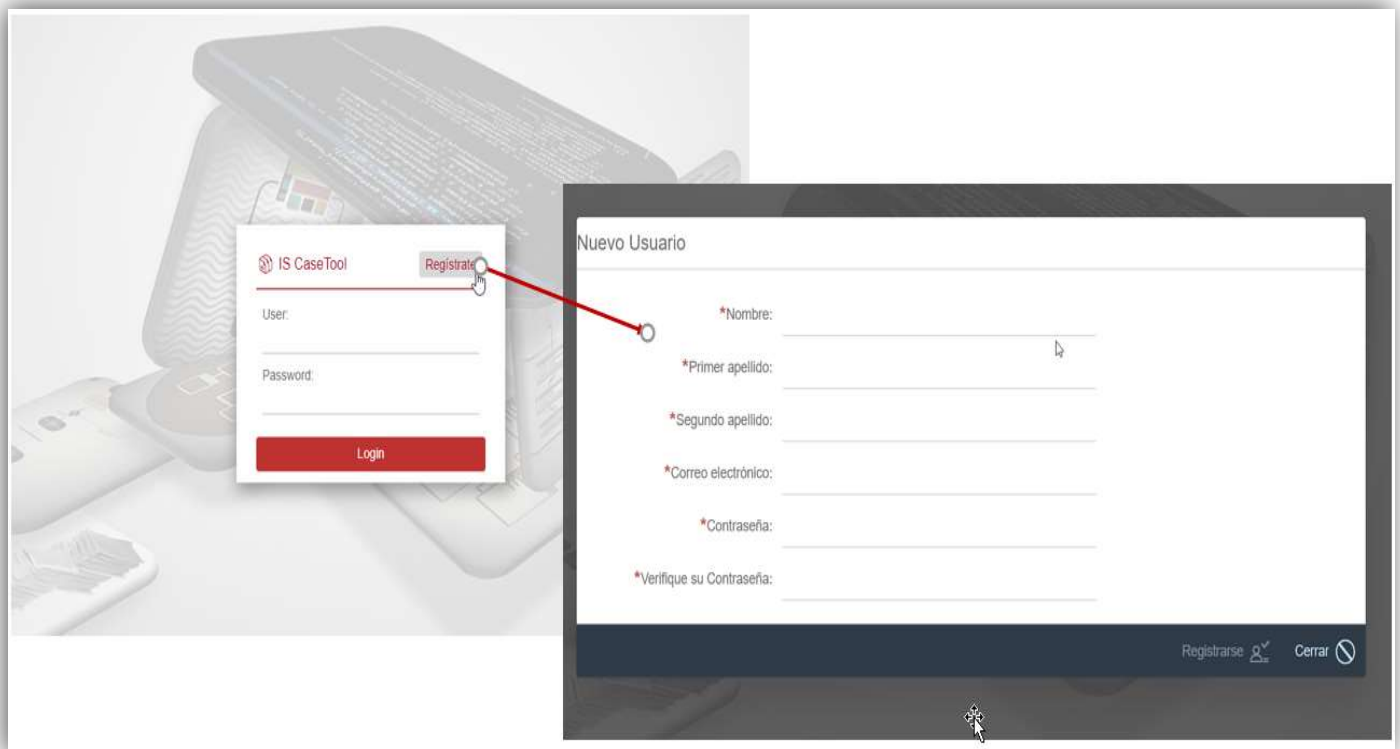


[Figura 61. Réplica de Figura 25.]

Para Miembro (No Administrador)

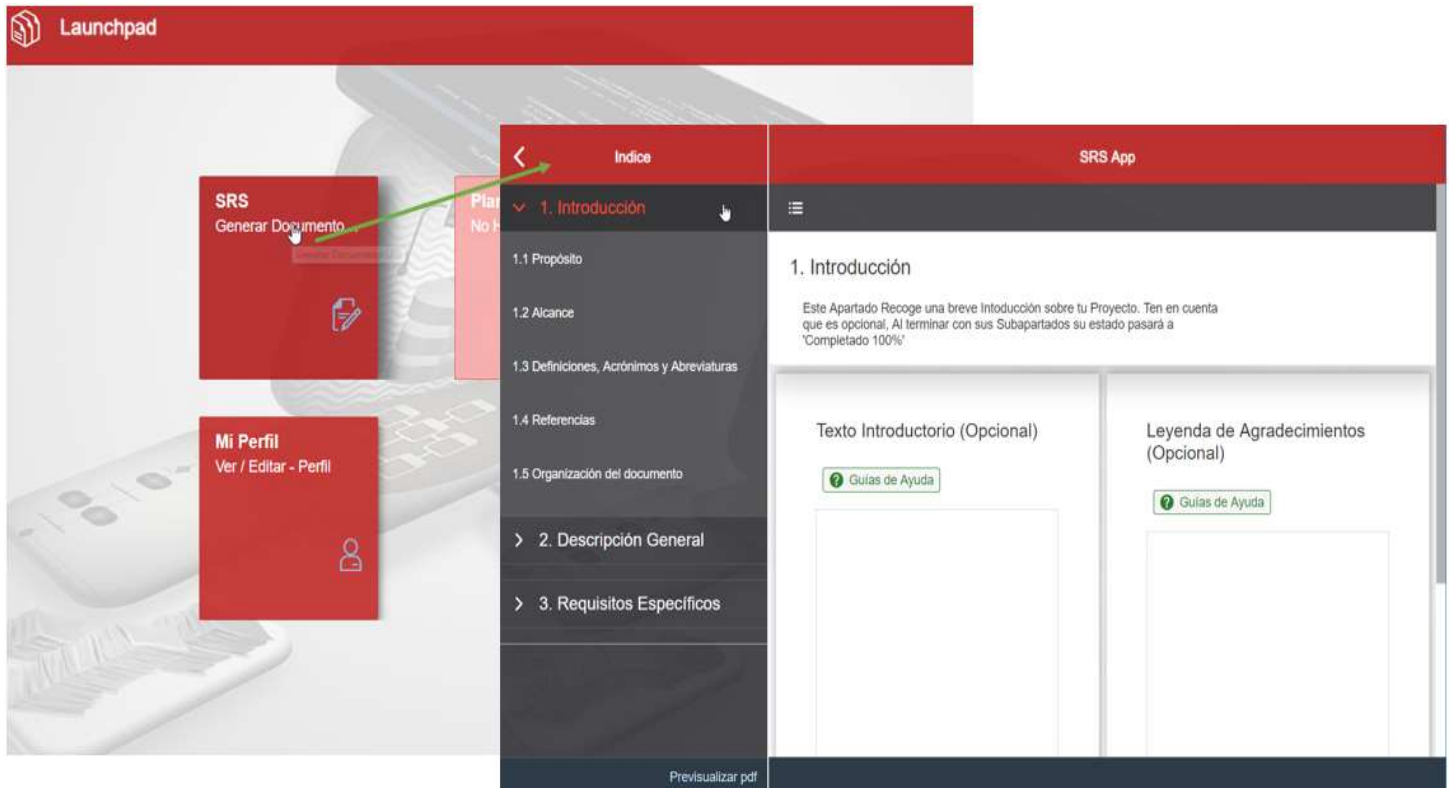
- **Registro de Usuario)**

- Acceder a la herramienta en la vista principal **“Login”**.
- Hacer clic en el botón **“Regístrate”**, se abrirá un formulario a rellenar.
- Rellenar los campos solicitados de forma coherente. De lo contrario no es posible realizar el Registro.
- Se activa el botón **“Registrarse”** al pie del formulario.
- Clic en el botón **“Registrarse”**. Una vez creado el Usuario en la BD, se muestra un mensaje de confirmación.



[Figura 62. Réplica de Figura 27.]

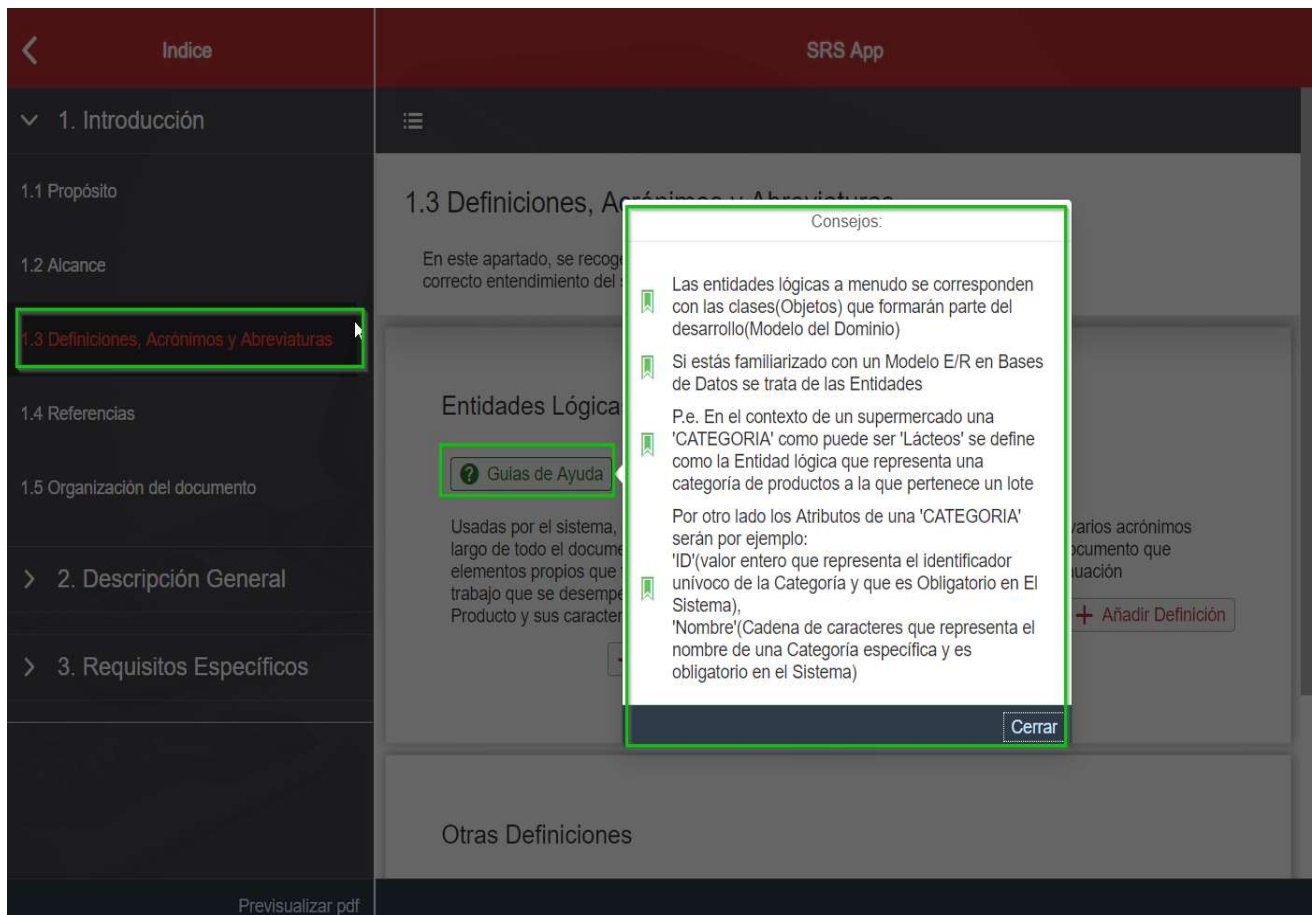
- **Acceso a un Documento concreto (SRS en este caso)**
 - Realizar un **“Login”** siendo Miembro No Administrador (ver Funcionalidades Generales)
 - Desde el catálogo **“Launchpad”** clic sobre el Tile **“SRS”**.



[Figura 63. Réplica de Figura 28.]

- **Visualización de Guías de Ayuda)**

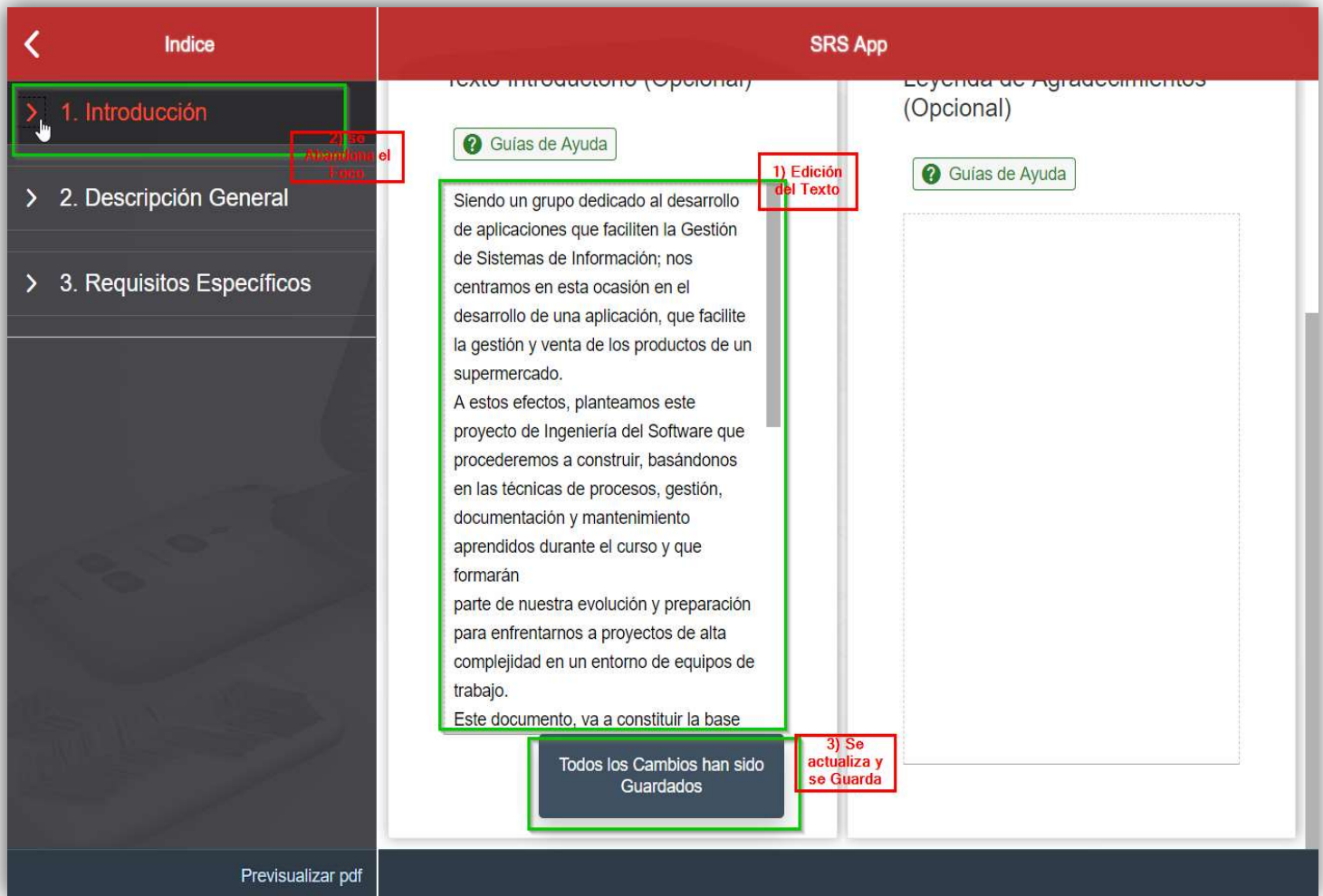
- Acceder al documento (SRS).
- Seleccionar cualquier apartado en el Menú de la Izquierda.
- A la derecha, clic sobre el botón **“Guías de Ayuda”**
- Se Muestran Consejos y Ejemplos para rellenar el apartado seleccionado.



[Figura 64. Réplica de Figura 30.]

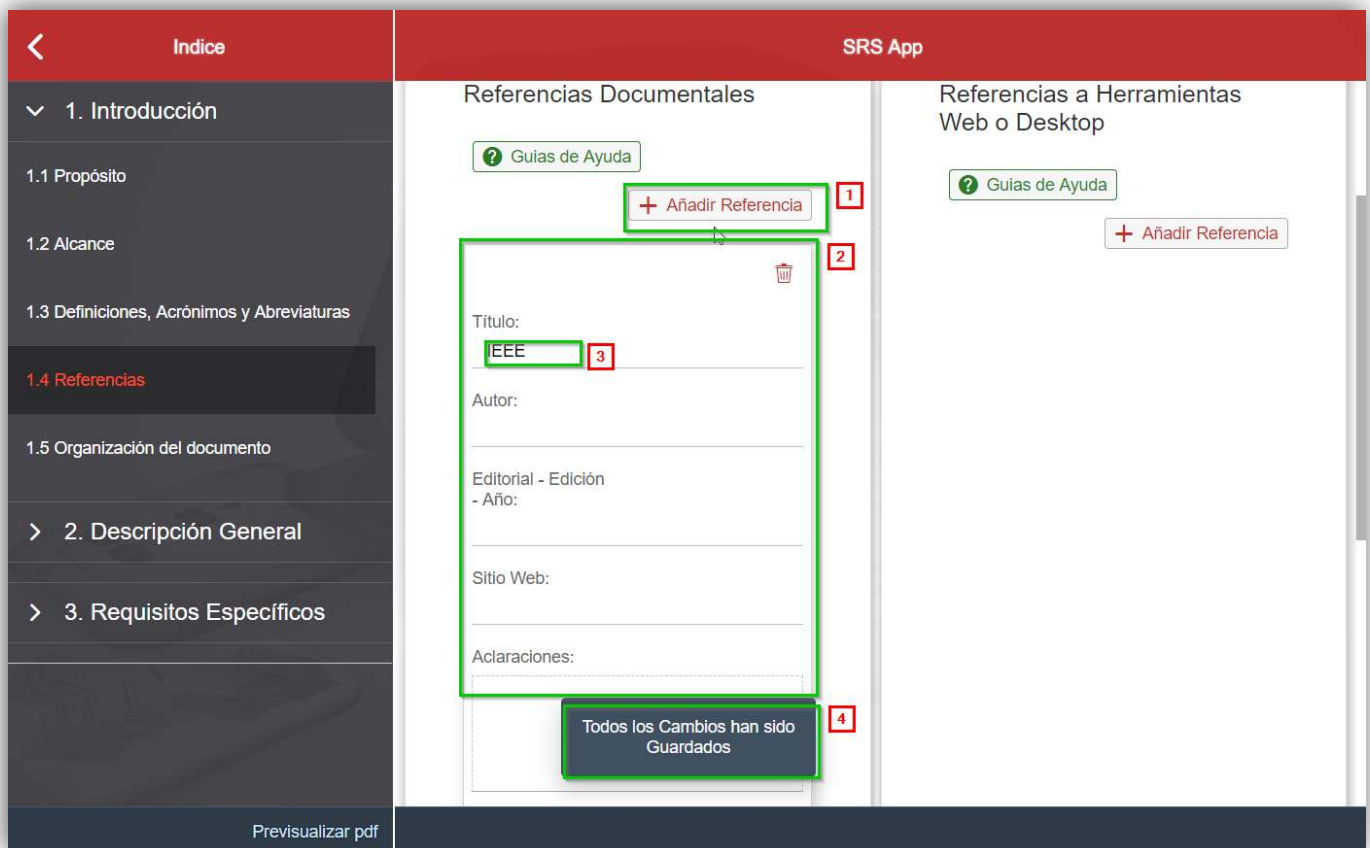
- **Edición de Apartados del Documento y Auto Guardado (TextArea)**

- Acceder al documento (SRS).
- Seleccionar cualquier apartado en el Menú de la Izquierda que presente un contenido con un Área de texto a Rellenar.
- Rellenar el área de texto.
- Clic fuera del área de texto, para que los datos se guarden automáticamente
- Se Recibe Mensaje de Autoguardado.



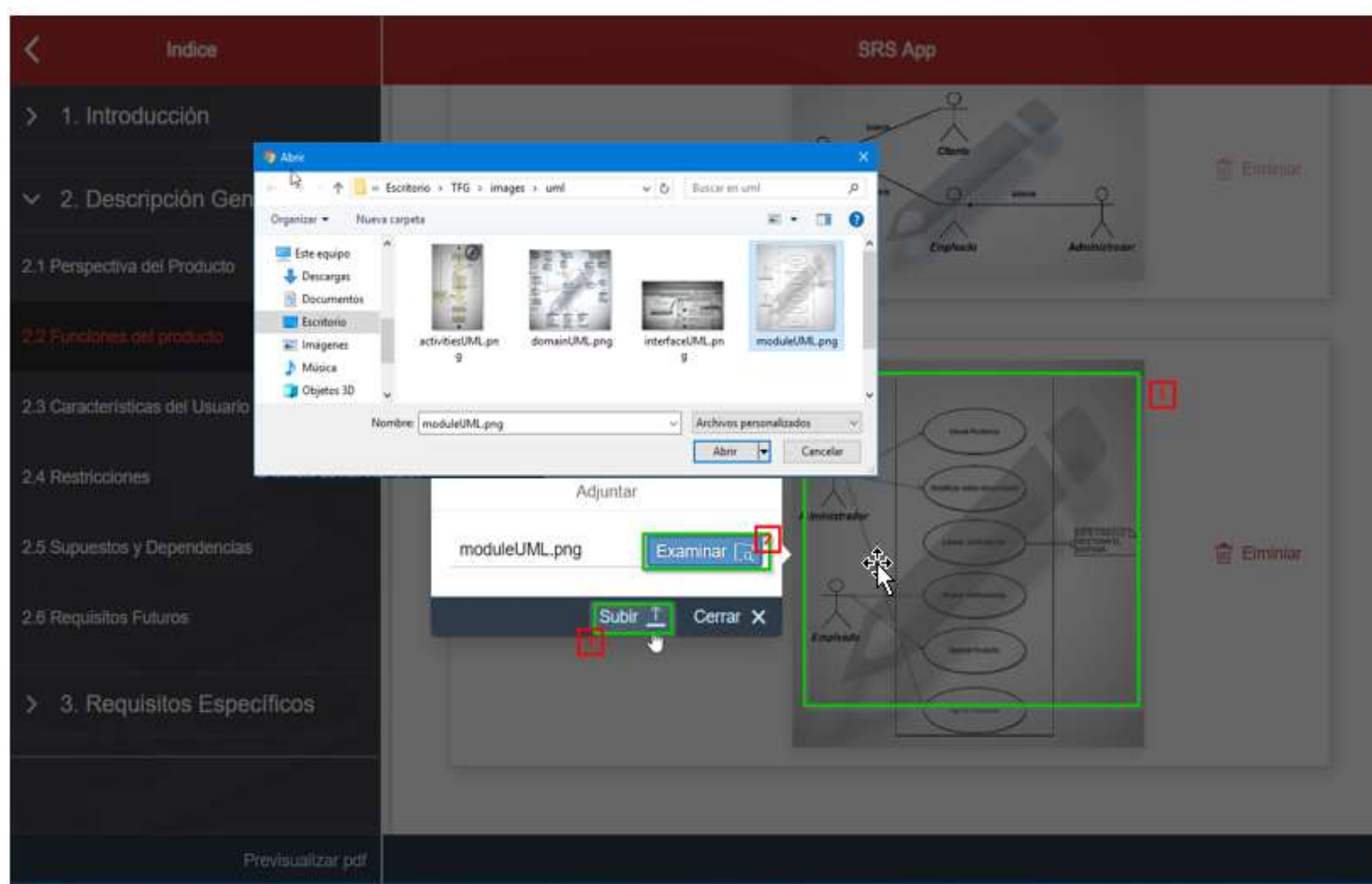
[Figura 65. Réplica de Figura 32.]

- **Edición de Apartados del Documento y Auto Guardado (Módulo Compuesto)**
 - Acceder al documento (SRS).
 - Seleccionar cualquier apartado en el Menú de la Izquierda que presente un contenido con un Módulo Compuesto por varios elementos a Rellenar.
 - Rellenar cualquier campo del módulo.
 - Clic fuera del campo, para que los datos se guarden automáticamente
 - Se Recibe Mensaje de Autoguardado por cada campo modificado.



[Figura 66. Réplica de Figura 33.]

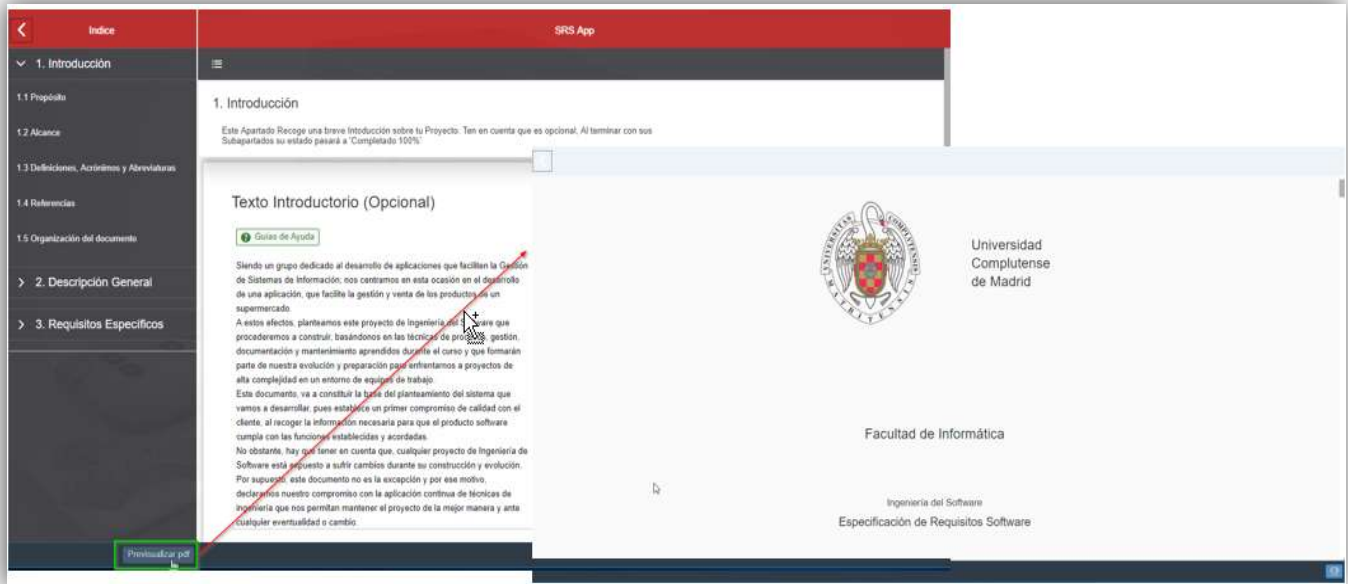
- **Edición de Apartados del Documento y Auto Guardado (Subir Imagen)**
 - Acceder al documento (SRS).
 - Seleccionar cualquier apartado en el Menú de la Izquierda que presente un contenido con una Imagen.
 - Clic sobre la Imagen.
 - Se abre un diálogo de selección.
 - Clic Sobre el botón **“Examinar”**
 - Se abre diálogo de selección de ficheros del Sistema Operativo.
 - Seleccionar una nueva Imagen.
 - Se rellena el diálogo con el nombre del fichero a subir.
 - Clic en el botón **“Subir”** ubicado al pie del diálogo.
 - La imagen se actualiza.
 - Se Recibe Mensaje de Autoguardado.



[Figura 67. Réplica de Figura 34.]

- **Previsualización del Documento Final)**

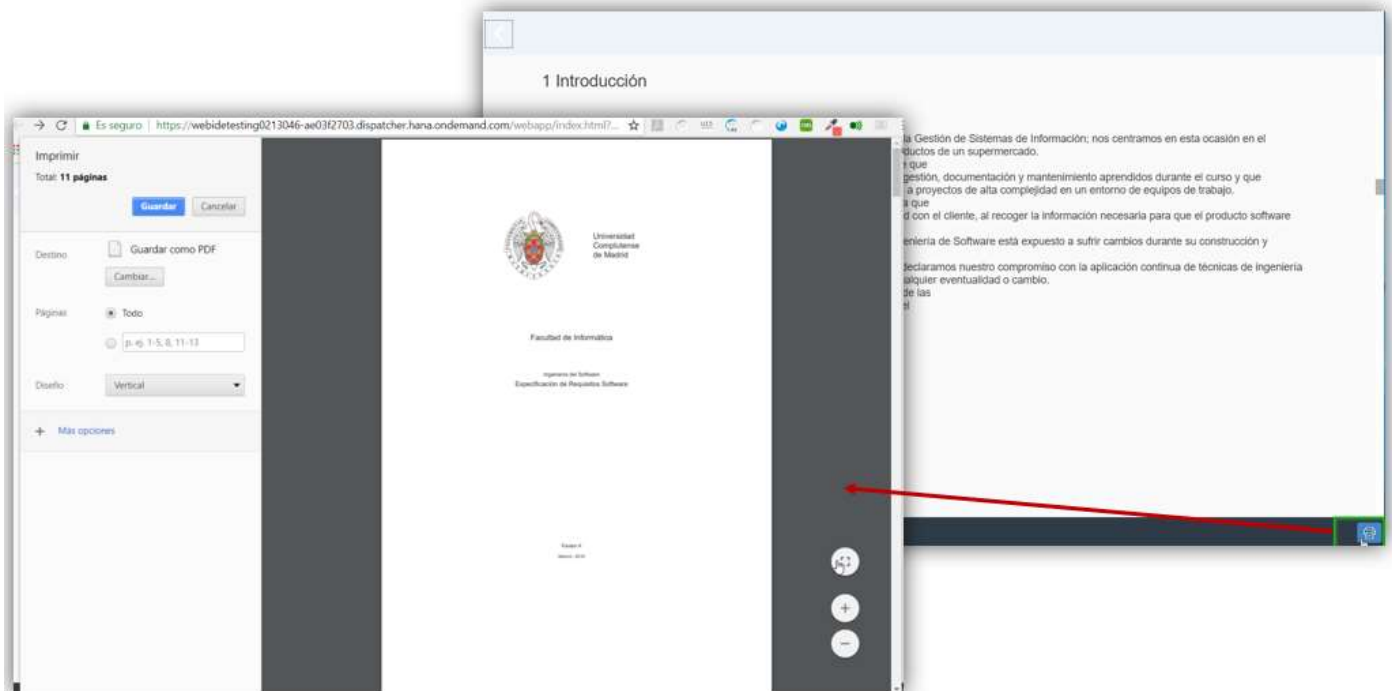
- Acceder al documento (SRS).
- Clic en el botón “**Previsualizar PDF**” ubicado al pie del menú de la izquierda (índice).
- Se muestra la previsualización del documento (SRS) que el equipo tiene hasta el momento sin llegar a generar un fichero PDF.



[Figura 68. Réplica de Figura 38.]

- **Impresión del Documento PDF)**

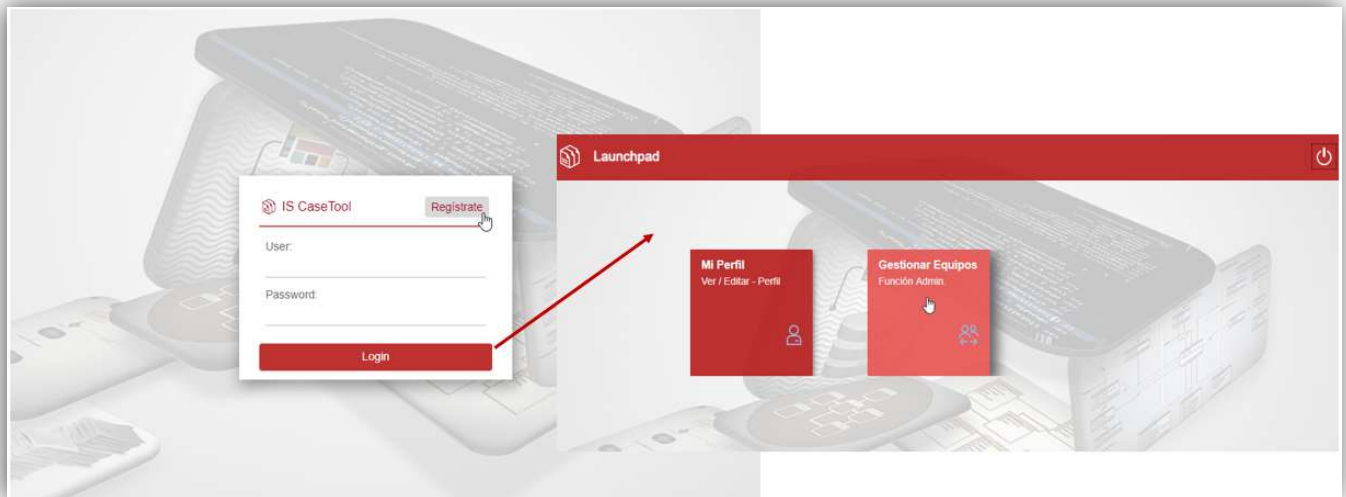
- Acceder a la vista de Previsualización PDF.
- Clic en el botón “**Imprimir**”.
- Se muestra el fichero PDF generado sobre la utilidad de impresión del propio Navegador.



[Figura 69. Réplica de Figura 40.]

Para Funcionalidades Generales (Todos Los Usuarios)

- **Login)**
 - Acceder a la herramienta, a través de Navegador Web
 - Se Muestra la Vista Principal con el formulario de autenticación
 - Rellenar el formulario, con los datos adecuados, de lo contrario dará un error en el siguiente paso.
 - Clic en el botón **“Login”**.
 - Se muestra el Catálogo Launchpad.



[Figura 70. Réplica de Figura 42]

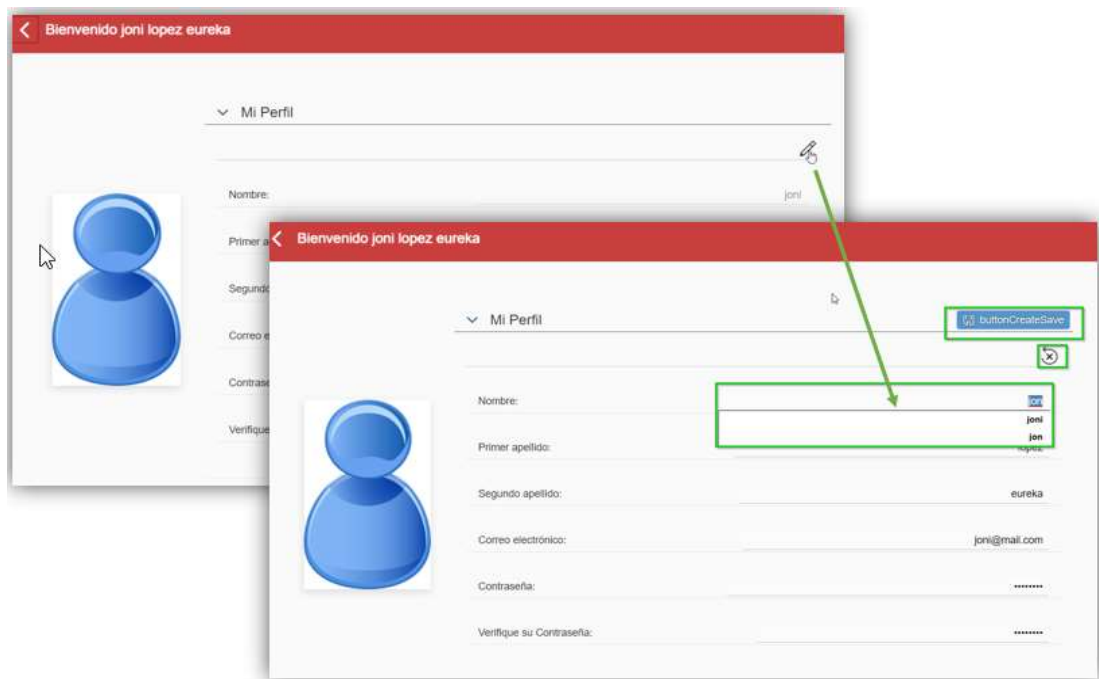
- **Acceso al Perfil de Usuario)**
 - Realizar un **“Login”** .
 - Desde el catálogo **“Launchpad”** clic sobre el Tile **“Mi Perfil”**.



[Figura 71. Réplica de Figura 44.]

- **Modificación de Datos del Usuario)**

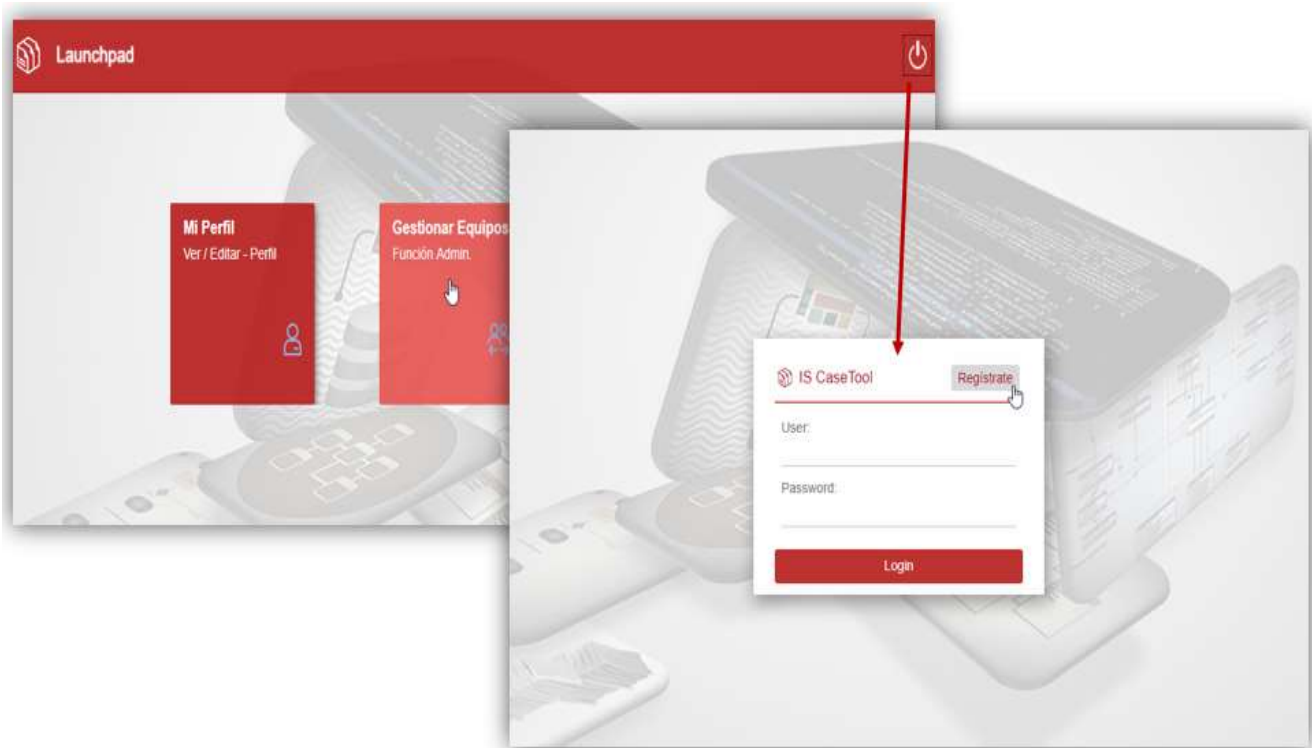
- Acceder a **“Mi Perfil”**
- Se muestra un formulario deshabilitado con los datos actuales.
- Clic sobre el botón **“Editar”** ubicado a la derecha sobre los datos.
- Se habilitan todos los campos del formulario y aparece el botón de **“Reset”** que sirve para cancelar la acción.
- Modificar los campos que lo requieran de forma coherente. De lo contrario no se puede continuar al paso siguiente.
- Se hace visible el botón **“Guardar”**.
- Clic Sobre el botón **“Guardar”**.
- Se recibe un mensaje de Confirmación de Cambios.
- Se muestran los datos actualizados y se deshabilita el formulario.



[Figura 72. Réplica de Figura 45.]

- **LogOut)**

- Acceder al LaunchPad.
- Clic en el Botón “**LogOut**”, ubicado en la cabecera a la derecha del catálogo LaunchPad.
- Se muestra la vista Principal “**Login**”.
- Otra opción es simplemente, cerrar la instancia del Navegador.



[Figura 73. Réplica de Figura 47.]